SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of Economics, Management and Statistics

PhD program in Economics, Statistics and Data Science - cycle XXXVII

Curriculum Big Data & Analytics for Business

Synthesis of Vector Space Models for Finance and Labour Market Analysis

Simone D'Amico

850369

Tutor: Prof. Matteo Pelagatti

Supervisor: Prof. Giancarlo Sperlì

Co-supervisor: Prof. Fabio Mercorio

Coordinator: Prof. Matteo Manera

ACADEMIC YEAR 2025/2026

Acknowledgments

Contents

List of Figures

List of Tables

Abstract

This doctoral thesis explores the application of Vector Space Models (VSMs) and modern language technologies to two high-impact domains: Labour Market Intelligence (LMI) and Financial Natural Language Processing (NLP). The research addresses domain-specific challenges through novel methodologies that advance beyond general-purpose language models, demonstrating how tailored computational approaches can extract meaningful insights from complex, specialized textual data.

In the European labour market context, this work tackles the critical problem of skill mismatch by developing frameworks for analyzing Online Job Advertisements (OJAs) across multiple languages. The thesis introduces: (i) KRAKEN, an unsupervised keyphrase extraction method achieving $F_1@5$ up to 24.4% and $F_1@10$ scores up to 28.6% on benchmark datasets and identifying emerging skills with 56.8–76.4% accuracy across five European languages; (ii) JobSet, a synthetic job advertisement dataset of 15,469 joc advertaisments that addresses data scarcity through strategic LLM generation, reducing perplexity (up to 26.7) and imporving skill explicitness compared to previous synthetic datasets; and (iii) VECTOR, a systematic framework for training, selecting, and aligning optimal word embedding models across 28 European countries, we generated 3,000+ and evaluate them with VECTORenabling comparable cross-national labour market analysis.

In the financial domain, the research confronts the transformative impact of social trading and retail investment platforms. Key contributions include: (i) the definition of the new task of Social Trading Action Detection (STAD) and the introduction of FINREDDIT-2K, a manually annotated dataset of 2,123 Reddit posts into three categories (buy, sell, or other), designed to serve as a benchmark for this task. We provide a benchmark with 57 models with the top three: Mistral-7B attains the highest F1-score (86.0%), followed by Neural-chat-7B (84.7%) and Phi-4-14B (84.6%). (ii) Novel approaches for assessing user reliability in financial social networks using heterogeneous graphs, with the proposed Content-based Centrality score outperforming traditional measures reaching 60% of according with expert judgments; and (iii) a systematic survey of multimodal models that integrates textual, numerical, and temporal data for enhanced financial forecasting.

The thesis establishes that domain-adapted vector space models consistently outperform general-purpose language models in specialized applications, while demonstrating the practical value of releasing curated datasets and open-source tools to foster reproducibility and collective progress. By bridging advanced NLP methodologies with real-world applications in labour economics and quantitative finance, this research provides both theoretical insights and practical frameworks for data-driven decision-making in increasingly complex information environments.

Thesis Outline

This thesis is organized into six coherent parts that systematically progress from foundational concepts to original research contributions across both application domains. The structure reflects the dual focus on Labour Market Intelligence and Financial NLP while maintaining methodological consistency through the application of Vector Space Models.

Part I: Introduction. This introductory part establishes the research landscape, presenting the motivation, challenges, and overarching contributions that frame the entire doctoral work.

Part II: Theoretical Background. Comprising Chapters 2 through 4, this part lays the theoretical foundation for the research. **Chapter 2** provides a comprehensive overview of Vector Space Models and Word Embeddings, tracing their evolution from frequency-based approaches to contemporary Large Language Models. **Chapter 3** examines Financial Natural Language Processing, focusing on domain-specific characteristics, data sources, and analytical tasks. **Chapter 4** explores Applications in Labour Market Intelligence, with particular emphasis on European labour market dynamics and the ESCO taxonomy framework.

Part III: Related Work. Spanning Chapters 5 to 7, this part critically reviews the existing literature. **Chapter 5** surveys foundational XAI frameworks and methodologies, establishing the theoretical groundwork for explainable vector space models. **Chapter 6** reviews related work in Finance, covering vector space models in financial prediction and social trend analysis. **Chapter 7** examines related work in LMI, including annotated job posting datasets and word embedding alignment techniques. .

Part IV: XAI for Vector Space Models. This part, consisting of **Chapter 8** presents a comprehensive survey of Explainable AI techniques for word embeddings, systematically categorizing existing methodologies into six distinct families and identifying critical research gaps in the interpretability of vector space models. This survey bridges the gap between complex embedding architectures and human-understandable explanations, providing a foundational framework for transparent semantic representations.

Part V: VSM for Labor Market. Containing Chapters 9 through 11, this part presents the core methodological contributions to Labour Market Intelligence. **Chapter 9** introduces `KRAKEN`, a novel semantic-based approach for keyphrase extraction from job advertisements. **Chapter 10** presents `JobSet`, a comprehensive synthetic job advertisement dataset generated through advanced language models. **Chapter 11** develops VECTOR, an extensive framework for training, selecting, and aligning optimal vector space models from online job ads across 28 European countries.

Part VI: VSM for Finance. Comprising Chapters 12 through 14, this part details the original contributions to Financial Natural Language Processing. **Chapter 12** evaluates the effectiveness of fine-tuning in financial NLP through the **Social Trading Action Detection (STAD)** task, introducing the

FINREDDIT-2Kdataset. **Chapter 13** proposes a contextual-based approach for enhancing user reliability using heterogeneous graphs in financial social networks. **Chapter 14** presents a comprehensive systematic survey of multimodal models for financial analysis, examining learning across diverse data modalities.

The thesis concludes with a synthesis of research findings, a discussion of limitations, and an identification of promising future research directions across both Labour Market Intelligence and Financial NLP domains.

List of Notation

Mathematical Notation

Symbol	Description
d	Document
D	Document collection
$f_{t,d}$	Frequency of term t in document d
$tf(t,d)$	Term frequency of t in d
$idf(t,D)$	Inverse document frequency of t in corpus D
$\vec{v}$, $\vec{v}_{word}$	Vector representation (embedding)
v_{size}	Dimensionality of word vectors
τ	Number of training epochs
α	Learning rate
ρ	Spearman's rank correlation coefficient
p_ρ	p-value of Spearman correlation
$IC(c)$	Information Content of concept c
$p(c)$	Probability of encountering concept c
N_c	Cardinality of concept c and its hyponyms
$HSS(w_1,w_2)$	Hierarchical Semantic Similarity between words w_1 and w_2
L	Lowest Common Ancestor (LCA) in taxonomy
th_{win}	Thresholds for window construction
th_{WW}	Thresholds for within-window score
kp_i	Keyphrase i
kp_d	Set of keyphrases for document d
$kp_i^{(2)}$	Set of all two-word combinations extracted from the words composing kp_i
WW_{kp_i}	Within-window score for keyphrase i
$BW_{kp_{i,d}}$	Between-window score for keyphrase i in document d
$P@k$, $R@k$, $F_1@k$	Precision, Recall, and F1-score at k
λ	Parameter of Poisson distribution for skill sampling
$sim(o_i,o_j)$	Cosine similarity between embeddings of objects i and j
Q, K, V	Query, Key, Value vectors in attention mechanism
d_k	Dimensionality of key vectors in attention
y_t, P_t, $r_{t+\Delta t}$	Stock prediction target, price at time t, return at $t+\Delta t$
G_s	Subgraph derived from subreddit s
$C_{G_s}(u)$	Centrality score for the user u in G_s

Acronyms and Abbreviations

Acronym	Description
VSM	Vector Space Model
NLP	Natural Language Processing
LLM	Large Language Model
LMI	Labour Market Intelligence
OJA	Online Job Advertisement
ESCO	European Skills, Competences, Qualifications and Occupations
HSS	Hierarchical Semantic Similarity
KPE	Keyphrase Extraction
STAD	Social Trading Action Detection
FSA	Financial Sentiment Analysis
CBOW	Continuous Bag of Words
SG	Skip-gram
BiLSTM	Bidirectional Long Short-Term Memory
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
GNN	Graph Neural Network
NER	Named Entity Recognition
BLI	Bilingual Lexicon Induction
CLS	Cross-Lingual Semantic fitting Score
WIH	Web Intelligence Hub
ISCO	International Standard Classification of Occupations
NACE	Statistical Classification of Economic Activities
XMLC	Extreme Multi-label Classification
XAI	Explainable Artificial Intelligence
IR	Information Ratio
IRR	Investment Return Ratio

Part I

Introduction

Chapter 1

Introduction

The exponential growth of digital textual data presents significant challenges for knowledge extraction, particularly in specialized domains where language carries critical business intelligence. In finance, textual sources like earnings reports, news articles, and social media contain signals about market movements and investor sentiment. Similarly, in labour market analysis, job advertisements and professional profiles reflect evolving skill demands and occupational trends.

The computational analysis of this textual data has become increasingly important for evidence-based decision making. However, the specialized nature of financial and labour market language, combined with its dynamic evolution and domain-specific semantics, requires tailored computational approaches that go beyond general-purpose language processing techniques.

This doctoral thesis is situated at the convergence of modern Natural Language Processing (NLP) and these two distinct yet data-rich application domains. It undertakes a comprehensive investigation into the potential of **Vector Space Models (VSMs)**—encompassing static word embeddings, contextualised representations, and the latest generation of **Large Language Models (LLMs)**—to decode complex, domain-specific narratives from heterogeneous textual data.

The utility of VSMs lies in their foundational ability to project discrete linguistic symbols into a continuous, geometric space. This transformation facilitates quantitative reasoning about qualitative concepts. Words, phrases, or entire documents, represented as vectors, can be compared, clustered, and operated upon using well-defined mathematical principles. The semantic relatedness between terms is captured by their proximity within this vector space, enabling tasks such as synonym detection, analogy resolution, and topic modelling. The evolution from count-based models like TF-IDF to predictive, dense embeddings such as Word2Vec [1] and GloVe [2] marked a significant leap, capturing latent semantic regularities. Subsequent advancements, notably the transformer architecture, gave rise to contextualised embeddings (e.g., BERT [3]) and generative LLMs (e.g., the GPT family), which dynamically adjust word representations based on their surrounding context, thereby effectively handling polysemy and complex syntactic structures.

This research is developed along two parallel, application-driven axes. The first axis focuses on the **European Labour Market**, a domain defined by its intrinsic multilingualism and the pressing need for real-time monitoring of skill supply and demand dynamics. The primary data source for this axis is **Online Job Advertisements (OJAs)**, which offer a timely and granular view of employer needs. The second axis delves into the **Financial Domain**, where market volatility is increasingly influenced by user-generated content on social media and investment forums. This necessitates models capable of moving beyond traditional sentiment analysis to detect specific trading intents and assess the reliability of information sources.

1.1 Research Domains

This thesis is organized around two primary application domains that share common methodological foundations in vector space modeling and natural language processing, yet address distinct real-world challenges. The research develops specialized approaches tailored to the specific characteristics and requirements of each domain.

1.1.1 Labour Market Intelligence: Challenges and Approaches

The analysis of European labour markets represents a critical application domain for advanced NLP techniques, characterized by its unprecedented scale, intrinsic multilingual nature, and profound socio-economic importance [4, 5, 6]. The digital transformation of recruitment processes has fundamentally altered labour market monitoring, making Online Job Advertisements (OJAs) a rich, real-time data source for understanding skill demands, occupational trends, and market dynamics across the European Union. Platforms such as the Web Intelligence Hub (WIH) [7] developed by Eurostat and Cedefop collected OJAs from more than 1000 sources in Europe since 2019, providing unprecedented granularity for analysing labour market phenomena.This data deluge enables researchers and policy makers to move beyond traditional, lagging indicators toward real-time assessment of emerging skills, regional disparities, and sectoral transformations (e.g. see results in [8, 9, 10, 11, 12]). The temporal sensitivity of OJAs allows for the detection of rapid market shifts, such as those induced by technological disruptions or economic crises, providing valuable early-warning signals for educational institutions and employment services.

However, the very richness of OJA data introduces significant computational challenges that this thesis addresses through novel NLP approaches. The heterogeneity of job advertisement formats, the informal and domain-specific language used in descriptions, and the varying quality of information across different platforms and countries require sophisticated text processing and normalization techniques. Furthermore, the dynamic nature of labour markets—where new occupations emerge and existing ones evolve—demands methods that can adapt to conceptual drift and terminological innovation without constant manual intervention.

In the context of Labour Market Intelligence, the rapid transformation of skills, driven by the digital and green transitions, demands agile tools to identify emerging competencies and skill mismatches. While OJAs are a rich resource, they present formidable challenges:

- **Linguistic and Terminological Variability**: The same occupation or skill can be described using a wide set of terminologies across different industries, companies, and European countries, complicating any form of standardised analysis. For instance, the role of a *Data Scientist* might be advertised as *Machine Learning Engineer*, *AI Specialist*, or use various language-specific equivalents, creating a semantic fragmentation that must be reconciled.

- **Scarcity of Annotated Data**: Supervised machine learning approaches for tasks like skill extraction and occupational classification are hampered by the scarcity of large-scale, high-quality labelled datasets. Manual annotation is prohibitively expensive, time-consuming, and difficult to scale across multiple languages, creating a significant bottleneck for the development of robust supervised models.

- **Multilingual and Cross-Country Comparability**: The European labour market is inherently multilingual. Developing models that perform consistently across different languages and that allow for a semantically meaningful comparison of skill demands in, for example, Italy versus Germany, requires cross-lingual alignment techniques beyond simple translation.

1.1.2 Financial Natural Language Processing: Challenges and Approaches

Financial Natural Language Processing represents a frontier where computational linguistics intersects with quantitative finance in increasingly complex and dynamic market environments [13, 14]. The digital revolution has fundamentally transformed how financial information is produced, consumed, and acted upon, creating both unprecedented opportunities and formidable challenges for automated analysis. The emergence of social trading platforms and the democratization of financial discourse have given rise to new forms of market behavior that traditional analytical frameworks struggle to comprehend and model effectively.

The landscape of financial communication has evolved from traditional institutional channels —earnings reports, analyst notes, and financial news—to include vast streams of user-generated content on platforms such as Reddit, X, and specialized investment forums. This transformation is not merely quantitative but qualitative: the language of finance has expanded to include informal discourse, memes, slang, and community-specific terminology that convey meaningful market signals alongside traditional financial jargon [15]. The GameStop short squeeze of 2021 serves as a paradigmatic example of how collective retail investor behavior, coordinated through social media, can disrupt established market dynamics and challenge conventional financial models [16].

The emergence of social finance has exposed significant challengers in financial NLP methodologies, particularly:

- **Beyond Sentiment: Detecting Trading Intent**: The sentiment of a social media post (positive or negative) is often orthogonal to the author's explicit trading intention (e.g., *buy*, *sell*, *hold*). A post expressing negative sentiment about a company's ethics might still conclude with an intention to *buy the dip*. This necessitates the development of more nuanced models for **Social Trading Action Detection (STAD)** that can discern specific action cues from general opinion.
- **User Reliability and Influence**: In the noisy ecosystem of financial forums, distinguishing credible signals from mere opinion or misinformation is paramount. This requires models that can assess user reliability by synthesising information from both network topology (e.g., user centrality, community structure) and the content of their contributions (e.g., consistency, historical accuracy), moving beyond simple engagement metrics.
- **Multimodal Integration for Forecasting**: Financial forecasting stands to benefit significantly from the integration of multiple data modalities, including historical price time series, textual news, and technical indicators. However, determining the most effective architectures and fusion strategies (early, intermediate, or late fusion) for combining these heterogeneous and temporally aligned data streams remains an open research question.

These domain-specific challenges underscore the necessity for tailored computational approaches. The following sections outlines the specific motivations and contributions made by this thesis to address these pressing issues in both Labour Market Intelligence and Financial NLP.

1.2 Research Motivation

In response to the critical challenges identified in the previous section, this thesis is motivated by several interconnected drivers that highlight the motivation of this research:

Scientific and Technological Motivation. While the literature features abundant applications of generic NLP, a methodological gap persists in adapting these models to highly specialized domains. The limitations of general-purpose models constitute a fundamental challenge in applied NLP. Pretrained architectures such as BERT or GPT, while effective on general corpora, consistently fail to

capture the nuanced lexicons characteristic of specialized domains. This is particularly evident in financial language with terms like *short squeeze* and *EBITDA*, and in labour market terminology such as *emerging digital skill* and *full-stack developer*. The inherent constraints of the *one-size-fits-all* paradigm become critically apparent when dealing with domain-specific polysemy, rapidly evolving terminologies, and specialized contextual dependencies. Furthermore, the absence of established methodologies for model selection in applied settings represents a significant research gap, leaving practitioners without systematic approaches to determine optimal embedding strategies for specific tasks and datasets. This technological shortcoming motivates the need for domain-adapted computational frameworks that can bridge the divide between theoretical advances and practical applications in specialized domains.

Applied and Impact Motivation. This research is strongly driven by concrete problems with significant socio-economic impact, addressing domain-specific challenges through novel computational approaches. For the Labour Market, this research tackles the critical challenge of identifying emerging skills by proposing a semi-automatic, data-driven methodology to detect novel competencies as they appear in Online Job Advertisements (OJAs). The inability to identify emerging competencies in real-time hampers effective policy-making and creates structural unemployment challenges. Simultaneously, in financial markets, the rise of social trading platforms has fundamentally transformed market dynamics, creating new forms of collective behavior and volatility that traditional analytical frameworks cannot adequately capture. The GameStop phenomenon exemplifies how retail investor communities can disrupt established market patterns, yet current tools lack the sophistication to analyze these complex social-financial interactions. These domain-specific challenges demand computational approaches that can provide timely, actionable insights for decision-makers in both labour policy and financial markets.

Data Resource Motivation. The advancement of specialized NLP applications is fundamentally constrained by critical data availability challenges. In labour market intelligence, the scarcity of large-scale, high-quality annotated job advertisements creates significant barriers for developing robust machine learning models, particularly for multilingual and cross-country analyses. Manual annotation processes are prohibitively expensive and cannot scale to meet the real-time demands of dynamic labour markets. Similarly, in financial NLP, the lack of expert-annotated social media data with precise trading intent labels limits the development of reliable models for social trading analysis. The absence of standardized, high-quality datasets in both domains not only impedes model development but also hinders reproducibility and comparative evaluation across different research efforts. These data scarcity issues motivate the need for innovative approaches to dataset creation and resource development that can support advanced computational analysis in specialized domains.

1.3 Contributions

Building upon the challenges identified in Section 1.1 and the research motivations outlined in Section 1.2, this thesis contributes to the advancement of data-driven analysis in two tightly connected research domains: *Labour Market Intelligence* and *Financial Natural Language Processing*. The core research objective of this work is to investigate how unstructured textual data, social signals, and heterogeneous information sources can be systematically transformed into actionable knowledge through representation learning, large-scale embedding models, and multimodal analysis. To this end, the thesis proposes methodological innovations, curated datasets, and reproducible frameworks that progressively address the limitations of existing approaches discussed in Sections 1.1 and 1.2.

The contributions are organized as a coherent research trajectory, where early chapters focus on foundational methods for textual knowledge extraction, subsequent chapters extend these methods

to large-scale labour market analysis, and the final part explores social and multimodal signals for financial decision-making. Specifically, this thesis makes the following primary contributions:

- **Embedding-based keyphrase extraction.** Addressing the challenge of extracting structured knowledge from unstructured text, this thesis introduces an unsupervised keyphrase extraction method that leverages word embeddings to identify semantically coherent phrases. The approach employs a context window expansion mechanism guided by semantic similarity metrics (cosine similarity and Pearson correlation), and is evaluated on multiple benchmark datasets, achieving state-of-the-art performance in performance@10 metrics. This contribution establishes the methodological foundation for subsequent applications and is presented in Chapter 9.

- **Synthetic data generation for labour market analysis.** To mitigate the scarcity and accessibility issues of high-quality labour market data, this thesis introduces `JobSet`, a dataset of synthetic yet realistic online job advertisements generated through the `JobGen` framework. By integrating ESCO taxonomy knowledge with real-world data distributions and iterative fitness evaluation, JobSet provides a scalable and reproducible resource for training and evaluating machine learning models in labour market applications. This contribution is detailed in Chapter 10.

- **Cross-country word embedding alignment for Labour Market Intelligence.** Tackling the challenge of multilingual and cross-country comparability in labour market studies, this thesis presents VEUCTOR, a systematic methodology for generating, evaluating, and aligning word embedding models across 28 European countries. The framework includes extensive hyperparameter optimization, cross-lingual alignment via the SeNSe technique, and the public release of optimized models, enabling reproducible and comparable analyses across linguistic contexts. This work is explored in Chapter 11.

- **Social trading action detection with large language models.** Moving toward the financial domain and social data analysis, this thesis introduces the novel task of *Social Trading Action Detection (STAD)*. The contribution comprises (i) FINREDDIT-2K, an expert-annotated dataset of social media posts labeled with explicit trading actions (buy, sell, other), and (ii) an extensive empirical evaluation of fine-tuned large language models on financial social media data. The study analyzes predictive performance, inference efficiency, and error patterns across multiple model families, establishing a benchmark for trading intent detection. This contribution is presented in Chapter 12.

- **Content-aware influence modeling in financial social networks.** To address the limitations of purely topological influence measures, this thesis introduces the *Content-based Centrality score (CbC)*, which integrates user engagement, sentiment extracted from posts, and classical network centrality measures. By incorporating semantic information into network analysis, the proposed metric enables more accurate identification of influential and trustworthy users in financial discussions. This methodology is described in Chapter 13.

- **A taxonomy and survey of multimodal financial forecasting models.** Finally, to systematize the transition from unimodal to multimodal financial modeling, this thesis provides a comprehensive taxonomy and comparative analysis of state-of-the-art multimodal approaches in financial forecasting. The survey examines input modalities, architectural choices, fusion strategies, and predictive tasks, highlighting open challenges and emerging research directions. This contribution is presented in Chapter 14.

Part II

Background

Chapter 2

Vector Space Models and Word Embeddings

Natural language is inherently ambiguous, complex, and unstructured, presenting significant challenges for computational analysis. To enable machines to interpret, compare, and process textual data, it is necessary to transform linguistic units—such as words, phrases, or entire documents—into a structured numerical format. This fundamental process, known as *vectorization*, lies at the core of modern Natural Language Processing (NLP). The primary goal of vectorization is to create representations that capture meaningful linguistic properties, so that the geometric relationships between these numerical vectors reflect the semantic relationships between the original text elements.

The *Vector Space Model (VSM)* provides the overarching mathematical framework for this paradigm. In a VSM, texts are represented as points (vectors) in a high-dimensional space, where each dimension typically corresponds to a specific feature of the language, such as a word or a character n-gram. The core premise is that the spatial arrangement of these vectors—their relative angles and distances—can encode linguistic relationships. This allows for the application of geometric and statistical methods to solve NLP tasks such as information retrieval, text classification, and semantic similarity analysis. The choice of how to define the dimensions of this space and how to populate the vectors with values distinguishes different families of approaches, which can be broadly categorized into *frequency-based* and *embedding-based* models. The following sections explore these two principal lineages of text representation.

2.1 Frequency-based Vector Space Model

The first generation of vector space models relied on sparse representations such as the Bag-of-Words (BoW) model and Term Frequency–Inverse Document Frequency (TF-IDF). In these approaches, each document or text unit is represented as a vector whose dimensions correspond to the terms of the vocabulary.

The Bag-of-Words model counts the frequency of each word in a text, disregarding word order and syntactic structure. While simple and effective for tasks like document retrieval, this approach leads to extremely high-dimensional and sparse vectors, where most entries are zero. Moreover, BoW fails to capture semantic similarity between words: the terms *car* and *automobile* are treated as completely unrelated, despite their close meaning.

TF-IDF improved upon BoW by weighting words according to their importance. It combines two components: the term frequency, which measures how often a word appears in a document, and the inverse document frequency, which reduces the weight of common terms across the corpus. Formally, the TF-IDF weight of a term t in a document d with respect to a collection of documents D is defined as:

$$\textit{tf-idf}(t,d,D) = \textit{tf}(t,d) \cdot \textit{idf}(t,D) \tag{2.1}$$

The *term frequency* is given by:

$$\textit{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.2}$$

where $f_{t,d}$ is the absolute frequency of term t in document d.

The *inverse document frequency* is defined as:

$$\textit{idf}(t,D) = \log\left(\frac{|D|}{1+|\{d \in D : t \in d\}|}\right) \tag{2.3}$$

where $|D|$ is the total number of documents in the collection, and $|\{d \in D : t \in d\}|$ denotes the number of documents in which the term t appears.

The TF reflects how often a word appears in a document, while the IDF penalizes words that are too common across the corpus (e.g., *the*, *and*). This weighting scheme enhanced information retrieval by emphasizing more informative terms. However, TF-IDF still inherits the fundamental limitations of BoW: vectors remain sparse, dimensionality grows with vocabulary size, and semantic relationships between words are ignored.

2.2 Distributional Semantics and Word Embeddings

The limitations of those models motivated the development of *distributed representations*, in which words are represented as dense vectors learned from data rather than fixed indices. The key intuition is that words occurring in similar contexts tend to have similar meanings. This idea, often referred to as the *distributional hypothesis*, underpins modern word embeddings.

The concept of word embeddings, fundamental to the study of distributional semantics, finds its origins in the *distributional hypothesis*, as proposed by Harris [17, 18]. This hypothesis asserts that linguistic elements can be identified based on their relative distributions within language: *x and y are included in the same element A if the distribution of x relative to the other elements B, C, etc. is in some sense the same as the distribution of y* [17]. In other words, the meaning of a word can be inferred from its distribution in the linguistic context. This principle suggests that similar words tend to appear in similar contexts, establishing an intrinsic connection between a word's position in a text and its meaning.

The application of the distributional hypothesis to textual corpora laid the groundwork for the emergence of *distributional semantics*, which aims to quantify and categorise the semantic properties of linguistic items based on their distributional patterns. According to [19], distributional semantics can be characterised by four key aspects: (i) it is a theoretical model to represent meaning, (ii) a computational framework to acquire meaning from language data, (iii) a practical methodology to construct semantic representations, and (iv) a cognitive hypothesis about the role of language usage in shaping meaning. In essence, distributional semantics seeks to represent the meaning of language through a formal model that encodes statistical distributions across contexts.

Building on the principles of distributional semantics, word embeddings can be defined as:

Definition 2.2.1 (Word Embedding)**.** *A word embedding is a dense, real-valued vector representation of a word, derived from a Distributional Semantic Model (DSM). It encodes the statistical distribution of the word across various linguistic contexts, capturing semantic and syntactic relationships with other words. Words with similar meanings are represented by vectors that are close in the embedding space, while words with different meanings are positioned further apart. Word embeddings thus provide a computationally efficient and expressive way to model the meaning of language, building on the principles of distributional semantics.*

This connection with distributional semantics provides word embeddings with powerful insights into the structure and meaning of language, advancing our capabilities in natural language understanding and processing. They represent a natural evolution from sparse vector space models, enabling more expressive and computationally efficient representations of textual information [20].

Word embeddings offer several key advantages over traditional sparse vector representations such as Bag-of-Words and TF-IDF:

- **Semantic Similarity:** Embeddings capture the semantic and syntactic relationships between words. Words with similar meanings are represented by vectors that are close in the embedding space, enabling tasks such as synonym detection, analogy reasoning, and semantic clustering.
- **Dimensionality Reduction:** Unlike sparse vectors that grow with vocabulary size, embeddings are dense and low-dimensional. This reduces memory requirements and computational complexity, making it feasible to process large corpora efficiently.
- **Generalization:** Word embeddings can generalize across contexts and domains. By learning from large corpora, embeddings can infer relationships between words even if they do not co-occur directly in the training data, improving performance in downstream NLP tasks.
- **Transferability:** Pre-trained embeddings can be reused across different tasks and domains, allowing models to leverage prior knowledge without requiring large labeled datasets for each new task.
- **Support for Complex Models:** Dense embeddings are well suited for integration with modern neural architectures, such as recurrent neural networks, transformers, and large language models, enabling more sophisticated natural language understanding and reasoning.

Overall, word embeddings provide a computationally efficient and semantically meaningful representation of language, bridging the gap between unstructured text and machine learning models. Additionally, embeddings allow for vector arithmetic and the use of distance or similarity functions, such as cosine similarity or Euclidean distance. This enables operations like analogical reasoning—for example, the classic relation *king* - *man* + *woman* $\approx$ *queen*—and more generally supports the quantitative analysis of semantic relationships and complex reasoning over words and concepts.

The main difference between traditional vector space models and word embeddings lies in their representation and expressiveness. While traditional sparse VSMs provide a general framework for mapping linguistic units into vectors, often relying on sparse, frequency-based counts, word embeddings are a specific type of VSM that learn dense, distributed representations from large corpora. By encoding the statistical distribution of words in context, embeddings capture semantic and syntactic relationships that sparse VSMs cannot. In other words, all word embeddings are instances of vector space models.

2.3 Static Word Embedding Models

Static word embeddings are representations in which each word type is mapped to a single fixed vector, regardless of the specific linguistic context in which it appears. These embeddings are learned in an unsupervised fashion from large text corpora, typically relying on the distributional hypothesis: words that occur in similar contexts should have similar meanings.

These models include Word2Vec [21], GloVe [2], and FastText [22] (among others). They proved highly valuable in many NLP tasks for capturing semantic and syntactic relationships, enabling analogies, clustering, and similarity-based reasoning. However, their fixed-vector nature limits their ability to account for multiple senses of a word or dynamic meaning according to context. In the following

subsections, we examine the major static embedding models, their training objectives, strengths, and limitations.

Word2Vec. A major breakthrough in word embeddings came with Word2Vec, introduced by Mikolov et al. [1, 21]. Unlike earlier count-based distributional models, Word2Vec uses predictive neural models to learn dense vector representations of words directly from large corpora.

Word2Vec learns word embeddings using a shallow neural network trained to predict word-context pairs from a large corpus. The network consists of an input layer representing the target or context word, a single hidden layer that generates the embeddings, and an output layer that produces a probability distribution over the vocabulary. Training maximizes the likelihood of observing actual word-context pairs, effectively capturing statistical regularities in word co-occurrences.

Within this framework, Word2Vec employs two main architectures: Continuous Bag-of-Words (CBOW) and Skip-gram (SG). CBOW predicts a target word based on its surrounding context words, averaging the embeddings of the context to generate a single representation. Skip-gram, conversely, predicts the surrounding context words given a target word, using the target's embedding to generate multiple output predictions for each context word. Both approaches result in dense vector representations that encode semantic and syntactic relationships between words.

In figure 2.1 we show the two models in comparison. The main advantage of using Word2Vec is that embeddings can be learned efficiently, with low complexity approaches, allowing larger embeddings to be learned from much larger corpora of text.

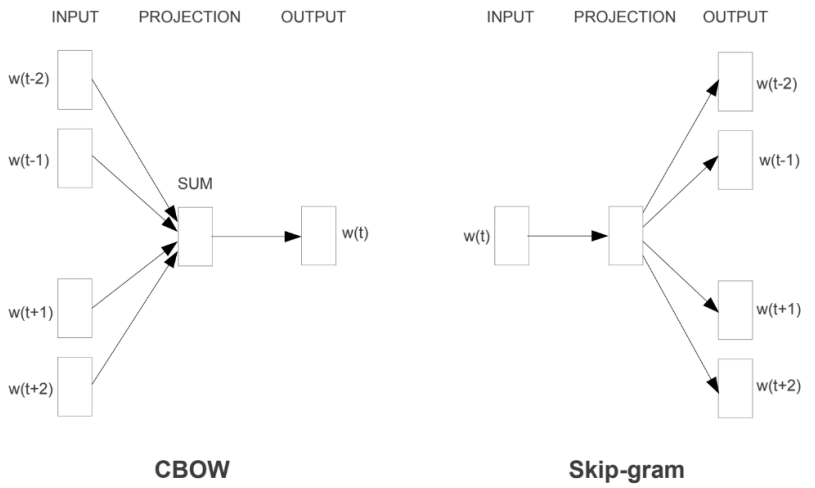


Figure 2.1: The model architectures from [1]. The CBOW architecture predicts the current word $w(t)$ based on the context $w(t-2),\ldots,w(t+2)$, and the Skip-gram predicts surrounding words given the current word.

CBOW is generally faster to train and produces smoother embeddings for frequent words, but it is less effective for representing rare words. Skip-gram, on the other hand, excels at learning precise embeddings for infrequent words, although it requires more computational resources due to predicting multiple context words for each target. Together, these architectures provide complementary strengths, enabling Word2Vec to efficiently generate high-quality word embeddings suitable for a wide range of natural language processing tasks.

Word2Vec also scales efficiently to very large corpora, making it practical for real-world NLP applications. However, Word2Vec has some limitations. It generates static embeddings, meaning that each word has a single vector representation regardless of context, which can be problematic for words with multiple senses. Additionally, it struggles with rare words and morphologically rich languages, since the model relies on observing sufficient occurrences of each token in the training data.

GloVe. GloVe (Global Vectors for Word Representation), introduced by Pennington et al. [2], is a word embedding model that combines the advantages of count-based and predictive approaches. Unlike Word2Vec, which learns embeddings by predicting local context words, GloVe constructs embeddings by factorizing a global word-word co-occurrence matrix derived from a large corpus. This allows the model to capture statistical information about how frequently words appear together across the entire dataset.

The training objective of GloVe is to learn word vectors such that their dot product approximates the logarithm of the probability of co-occurrence. Specifically, for words i and j, the model minimizes a weighted least squares loss function that penalizes the difference between the predicted and actual log co-occurrence counts. This formulation ensures that both frequent and rare co-occurrences contribute meaningfully to the learned embeddings.

GloVe embeddings are dense, low-dimensional vectors that encode semantic and syntactic relationships. One of the main strengths of GloVe is its ability to incorporate global corpus statistics, which often results in more stable and accurate representations than purely local prediction-based models. Additionally, GloVe preserves linear substructures that facilitate analogical reasoning, similar to Word2Vec. However, like Word2Vec, GloVe generates static embeddings, meaning that each word has a single vector regardless of context, and it may struggle with rare or morphologically complex words.

FastText. FastText, developed by Bojanowski et al. [22], extends the Word2Vec architecture by incorporating subword information into the representation of words. While Word2Vec and GloVe treat each word as an atomic unit, FastText represents words as a collection of character n-grams. Each word embedding is thus obtained by summing or averaging the embeddings of its constituent subword units. This design enables the model to capture morphological features and share information across words with similar forms.

During training, FastText uses the same predictive objective as the Skip-gram model, but instead of learning vectors only for entire words, it learns vectors for all character n-grams that compose them. For example, the word *financial* would be represented not only by its full word vector but also by subcomponents such as *fin*, *inan*, and *nancial*. This approach allows FastText to generate meaningful embeddings for rare or even unseen words by composing them from their subword parts, a significant improvement over previous models that relied solely on observed word occurrences.

FastText offers several advantages. It effectively handles morphologically rich languages and reduces the problem of out-of-vocabulary (OOV) words, providing better generalization on unseen data. The inclusion of subword information also enhances the model's ability to capture semantic nuances related to prefixes, suffixes, and word roots. However, FastText still produces static embeddings, meaning that it cannot distinguish between different senses of the same word depending on context. Moreover, the inclusion of subword representations increases computational complexity and memory requirements compared to standard Word2Vec.

2.4 Contextual Word Embedding Models

While static word embeddings have proven extremely useful, they exhibit inherent limitations due to their fixed-vector nature. Specifically, a single vector is assigned to each word type, regardless of the particular sense or usage that word has in a sentence. This causes problems in handling polysemy (a word having multiple meanings) and contextual variation: for example, the word *bank* in *financial bank* vs. *river bank* will have the same static embedding, even though its meaning is quite different.

Another limitation is that static embeddings fail to adapt to nuanced syntactic or semantic cues that depend on the sentence-level or discourse-level context. They are also less effective in tasks where precise interpretation of a word depends heavily on its neighbor words, or where long-range

dependencies matter.

To overcome these problems, the class of contextual word embeddings models emerged, which generate different embeddings for each token occurrence, taking into account its surrounding context. These models are generally based on neural architectures (such as LSTMs or Transformers) trained with language modelling objectives or other self-supervised tasks. Contextual embeddings can disambiguate word senses, adjust representation depending on usage, and better capture subtle meaning shifts.

In the following subsections, we explore representative contextual embedding models, beginning with ELMo [23], and then moving to Transformer-based models such as BERT [3], describing their architectures, training, strengths, and weaknesses.

ELMo. The first significant step toward contextual word representations was achieved with ELMo (Embeddings from Language Models) proposed by Peters et al. [23]. Unlike static word embeddings, ELMo produces context-dependent representations by generating a different vector for each token occurrence, depending on its linguistic environment within a sentence.

ELMo is based on a deep bidirectional language model (biLM) that uses two layers of Long Short-Term Memory (LSTM) networks. The forward LSTM processes the sentence from left to right, while the backward LSTM processes it from right to left. This bidirectional structure enables the model to capture both past and future contextual information for each word. The hidden states from both directions are concatenated and linearly combined to form the final embedding for each token.

Training is performed in a self-supervised manner using a language modelling objective. The model is trained to predict the next word given its previous context in the forward direction, and to predict the previous word given its subsequent context in the backward direction. Once trained, the internal representations of the biLM are used as contextual embeddings. These representations can be incorporated into downstream NLP tasks by concatenation or fine-tuning, allowing ELMo to improve performance across a wide range of applications such as question answering, sentiment analysis, and named entity recognition.

ELMo embeddings are dynamic: the same word will have different representations depending on its sentence context. This ability to disambiguate polysemous words marked a major advancement over static embeddings. However, the sequential nature of LSTM-based architectures makes ELMo computationally expensive and less effective in capturing very long-range dependencies compared to later Transformer-based models such as BERT.

BERT. BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin et al. [3], represents a major leap forward in embedding models by producing deep contextualized word representations using the Transformer architecture. Whereas previous models (e.g. Word2Vec, GloVe, FastText) assign each word a single, static vector, BERT generates different embeddings for each occurrence of a token, informed by its full left and right context within a sentence or sentence pair.

BERT's architecture is built as a stack of Transformer encoder layers [24]. Each encoder layer consists of two main sub-layers: a **multi-head self-attention** mechanism followed by a position-wise fully connected feed-forward network. Residual connections and layer normalization are applied around each of these sub-layers as shown in Fig. 2.2.

Self-attention is at the core of BERT. For each token in the input, the model computes three vectors: **Query** (Q), **Key** (K), and **Value** (V) via learned linear transformations of the token's embedding. The attention weight from token i to token j is computed as:

$$\text{Attention}(Q_i, K, V) = \text{softmax}\left(\frac{Q_i K_j^T}{\sqrt{d_k}}\right) V_j \tag{2.4}$$

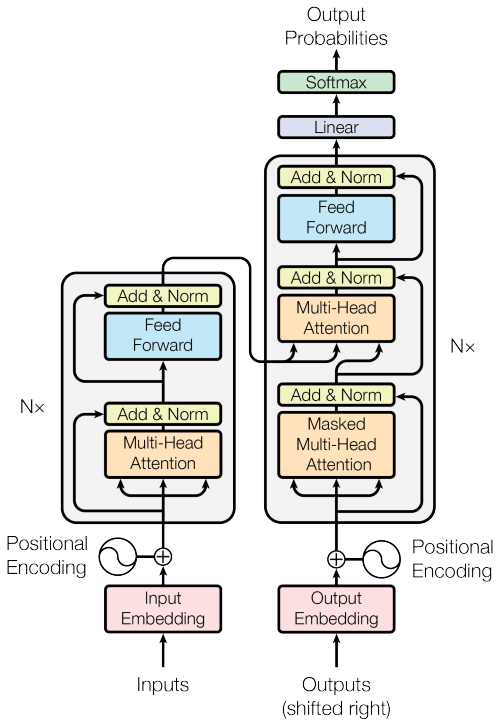


Figure 2.2: The encoder-decoder architecture of the Transformer model from [24], featuring multi-head attention mechanisms and feed-forward networks with residual connections and layer normalization.

where d_k is the dimensionality of the key vectors and acts as a scaling factor. In **multi-head attention**, multiple such attention mechanisms operate in parallel (heads), each with its own Q, K, V projections; their outputs are concatenated and linearly transformed. This allows the model to attend to different types of relationships among tokens.

BERT offers several important advantages over previous embedding models. By producing context-dependent embeddings for each token occurrence, it can effectively disambiguate polysemous words and capture subtle semantic nuances. Its deep bidirectional architecture enables rich context modelling, as each representation integrates information from both left and right contexts across multiple layers. Another major strength lies in its transferability: once pre-trained, BERT can be fine-tuned for a wide range of downstream NLP tasks—such as classification, question answering, and named entity recognition—often achieving state-of-the-art results. Moreover, the self-attention mechanism allows BERT to capture long-range dependencies more effectively than recurrent models like RNNs or LSTMs.

Nevertheless, BERT presents some limitations. Its training process is computationally expensive and requires large-scale data and hardware resources, while inference remains slow for long sequences. Although BERT successfully models contextual variation, its performance may decrease when the context is minimal or ambiguous.

In summary, contextual word embedding models represent a significant advancement over static embeddings by generating representations that vary according to the specific linguistic context of each token. Models such as ELMo and BERT leverage neural architectures—LSTMs for ELMo and Transformers with self-attention for BERT—to capture both local and long-range dependencies, disambiguate polysemous words, and provide rich, transferable embeddings suitable for a wide range of downstream tasks. While these models require substantial computational resources and careful pre-training, their ability to encode deep contextual information marks a major step forward in natural

language understanding. This development sets the stage for Large Language Models, which extend the principles of contextual embeddings to much larger architectures, enabling more powerful and flexible language representations capable of complex reasoning and generation.

2.5 Large Language Models

Large Language Models (LLMs) represent the latest evolution of word and token embedding models, building on the principles of contextual embeddings while significantly extending their scale, capability, and applications. While models such as ELMo and BERT produce embeddings that vary with context, LLMs are distinguished by their enormous number of parameters, deep Transformer-based architectures, and ability to perform complex language understanding and generation tasks across diverse domains.

It is meaningful to treat LLMs as a separate category within word embeddings because they are not merely larger versions of previous models. Their scale enables emergent properties that go beyond token-level context modeling, such as few-shot learning, reasoning, and the generation of coherent multi-sentence outputs. Furthermore, LLMs often incorporate additional training strategies, such as causal language modeling or mixture-of-experts architectures, which enhance generalization and adaptability.

Despite these advances, LLMs retain the core principle of contextual embeddings: each token is represented as a vector that depends on its surrounding context. This connection allows LLMs to inherit the advantages of contextualized representations—disambiguation of polysemy, deep semantic understanding, and transferability—while extending them to more complex, large-scale tasks in natural language processing. In the following sections, we describe the main architectural principles, pretraining methods, and characteristics of contemporary LLMs, highlighting how they build upon and surpass previous contextual models.

To illustrate the diversity and capabilities of Large Language Models, we next examine some of the most prominent families of LLMs. For each family, we highlight key architectural choices, pretraining strategies, and distinctive features that differentiate them from one another, while demonstrating how they build upon the principles of contextual embeddings.

GPT (Generative Pre-trained Transformer). OpenAI developed the GPT family[1] [25, 26] represents a groundbreaking line of LLMs. These models made use of Transformer-based decoders for generative language modelling. The first GPT model, introduced in 2018, demonstrated that pretraining a unidirectional Transformer on large-scale text corpora, followed by fine-tuning on specific tasks, could achieve strong performance across multiple natural language processing applications. This approach leveraged the causal language modeling objective, where the model predicts the next token given all previous tokens, enabling coherent text generation.

Over successive generations, GPT models have grown substantially in both scale and capability. GPT-2, released in 2019, expanded the model size to 1.5 billion parameters and demonstrated unprecedented text generation quality, capable of producing coherent multi-paragraph outputs. GPT-3, introduced in 2020 with 175 billion parameters, showcased remarkable few-shot and zero-shot learning abilities, allowing the model to perform new tasks with minimal or no task-specific fine-tuning. GPT-4 further improved performance, incorporating multimodal inputs, enhanced reasoning capabilities, and a more robust understanding of nuanced language.

Across all versions, the GPT models share key characteristics: they employ a deep stack of Transformer decoder layers with multi-head self-attention, use large-scale unsupervised pretraining on diverse corpora, and generate context-sensitive embeddings for each token. Their strengths lie in fluent text generation, adaptability to new tasks via few-shot learning, and emergent reasoning capabilities

[1] `https://openai.com/`

that appear at scale. The evolution of GPT has set a benchmark for generative LLMs, influencing the design and development of many subsequent models in both research and industry.

LLaMA. LLaMA[2] (Large Language Model Meta AI) [27, 28], developed by Meta, represents a family of Large Language Models designed to provide high performance while improving accessibility and efficiency for the research community. Introduced in 2023, LLaMA models are based on Transformer decoder architectures, similar to GPT, but optimized to achieve competitive performance with fewer parameters. This design choice allows researchers to experiment with large-scale language models without the extensive computational resources required by the largest GPT models.

The LLaMA family includes models of varying sizes, ranging from tens of billions to hundreds of billions of parameters. Pretraining is performed on diverse, high-quality text corpora using causal language modeling, enabling the models to generate coherent and contextually relevant text. Despite their smaller scale compared to the largest GPT models, LLaMA models demonstrate strong few-shot and zero-shot capabilities, effective reasoning, and the ability to produce fluent and accurate outputs across a variety of NLP tasks.

Key characteristics of LLaMA include efficient parameter usage, a focus on accessibility for academic research, and adaptability across multiple downstream tasks. By balancing model size, training efficiency, and performance, LLaMA provides an important alternative to larger, more resource-intensive LLMs, making it a widely adopted choice for both experimentation and practical applications in natural language understanding and generation.

Mistral. The Mistral family[3], developed by Mistral AI, represents a new generation of open-weight Large Language Models designed to maximize efficiency and performance through architectural innovations and optimized training strategies. Introduced in late 2023, Mistral models are fully based on the Transformer decoder architecture and are trained with causal language modeling objectives. However, they incorporate several improvements over previous models, including sliding window attention mechanisms and optimized positional encodings, which enhance their ability to process longer contexts with lower computational costs.

The first release, Mistral 7B, demonstrated that a well-trained model with only seven billion parameters could outperform much larger models, such as LLaMA-2 13B, across multiple benchmarks. This efficiency stems from refined data curation, high-quality pretraining corpora, and the use of grouped-query attention (GQA), which reduces the memory footprint while maintaining strong performance. Later variants, including Mixtral (a mixture-of-experts architecture), introduced sparse activation strategies that allow only a subset of parameters to be used at inference time, significantly improving scalability and cost-efficiency.

Mistral models emphasize open availability, reproducibility, and efficiency, making them a strong alternative to proprietary models. Their design philosophy focuses on providing state-of-the-art performance while keeping models lightweight and accessible for research and commercial use. This balance between compactness and capability positions the Mistral family as a major step toward democratizing access to powerful large language models.

DeepSeek. DeepSeek represents a recent and highly influential addition to the landscape of Large Language Models, developed by the Chinese research organization DeepSeek AI[4]. Emerging in 2024, the DeepSeek family of models is characterized by its exceptional efficiency and cost-effectiveness, demonstrating that large-scale language modeling can achieve state-of-the-art performance without

[2] https://www.llama.com/

[3] https://mistral.ai/

[4] https://www.deepseek.com/en

relying on massive computational resources. DeepSeek adopts a decoder-only Transformer architecture, similar to GPT, but introduces a series of engineering and training optimizations that drastically reduce training costs and energy consumption.

One of the key innovations of DeepSeek lies in its optimized training pipeline, which combines a highly efficient distributed training framework with advanced quantization and sparsity techniques. This enables DeepSeek models to maintain competitive accuracy and reasoning ability even when trained with significantly fewer GPU hours compared to Western counterparts such as GPT-4 or Claude. Furthermore, DeepSeek incorporates sophisticated fine-tuning and instruction-following mechanisms that improve its contextual understanding and alignment with user intent.

The release of DeepSeek also has strategic significance for the global AI ecosystem. By open-sourcing portions of its architecture and emphasizing efficient, scalable model development, DeepSeek has contributed to the democratization of advanced language models, fostering broader participation in LLM research. Its success underscores a shift in focus within the field—from scaling parameters to optimizing training methodologies and deployment efficiency—paving the way for a new generation of sustainable, high-performance LLMs.

Other Notable Families of LLMs. Beyond the major GPT, LLaMA, and Mistral families, several other large language models have emerged, each contributing unique innovations and design philosophies to the LLM landscape. Among these, **Gemma**, developed by Google DeepMind[5], is a family of lightweight, open-weight models built on the same research foundations as Gemini. Gemma is optimized for efficiency and responsible AI deployment, focusing on transparency, reduced computational costs, and strong performance in multilingual and reasoning tasks. Its compact architecture makes it particularly suited for academic and edge applications where resource constraints are critical.

Gemini, also developed by Google DeepMind, represents a new generation of multimodal LLMs, succeeding previous models such as LaMDA and PaLM. Gemini models natively support text, image, and audio modalities, and leverage advanced techniques like Multi-Query Attention and Mixture-of-Experts for efficient large-context processing. Certain variants, such as Gemini 1.5 Pro, can handle contexts up to 1 million tokens, enabling complex document understanding and extended conversations. Gemini is designed for multilingual capabilities, supporting over 40 languages.

Claude, introduced by Anthropic[6], is designed around the principle of *constitutional AI*, emphasizing safety, alignment, and controllability in large-scale language models. Instead of relying solely on reinforcement learning from human feedback (RLHF), Claude models integrate a set of guiding principles—"a constitution"—to steer behavior during fine-tuning, thereby improving model interpretability and ethical robustness. Claude has been particularly successful in dialogue-based applications, maintaining coherence and factual consistency across long conversational contexts.

Falcon, developed by the Technology Innovation Institute (TII)[7], represents another major open-weight initiative. Trained on massive, high-quality datasets with a strong emphasis on efficiency and reproducibility, Falcon models achieve competitive performance while remaining openly accessible for research and industry. Their design highlights the growing trend toward transparent, community-driven LLM development.

Finally, the **Phi** family[8], introduced by Microsoft, explores the idea that high-quality synthetic data and compact architectures can yield models that perform comparably to much larger systems. The Phi models, particularly Phi-2 and Phi-3, demonstrate impressive reasoning and language understanding capabilities despite their relatively small parameter count, offering a proof of concept for scaling down large language models without significantly sacrificing performance.

[5] https://deepmind.google/

[6] https://www.anthropic.com/

[7] https://www.tii.ae/

[8] https://azure.microsoft.com/en-us/products/phi

Table 2.1: Technical specifications and release timeline of prominent large language models, highlighting architectural differences, scaling properties, and multilingual capabilities across industry and open-source developments.

Model / Family	Developer	Parameters	Context Length	Multilingual Support	Release Date	Open/Closed	Key Features
GPT-1	OpenAI	120M	512	No (English only)	2018-06	Open	First GPT model, transformer decoder
GPT-2	OpenAI	1.5B	1024	No (English only)	2019-02	Open	Improved text generation capabilities
GPT-3	OpenAI	175B	4096	Limited (Basic multilingual)	2020-05	Closed	Few-shot learning, large scale
GPT-3.5	OpenAI	175B	16384	Yes (Strong multilingual)	2022-11	Closed	Instruction tuning, ChatGPT
GPT-4	OpenAI	1.76T*	128K	Yes (Excellent across languages)	2023-03	Closed	Multimodal, improved reasoning
GPT-4 Turbo	OpenAI	-	128K	Yes (Excellent across languages)	2023-11	Closed	Extended context, cheaper
GPT-4o	OpenAI	-	128K	Yes (State-of-the-art multilingual)	2024-05	Closed	Omnimodal, real-time audio/vision
Gemini Pro	Google	-	32K	Yes (Excellent multilingual)	2023-12	Closed	Multimodal native, text+images
Gemini Ultra	Google	-	32K	Yes (Excellent multilingual)	2024-02	Closed	Top-tier performance, advanced reasoning
Gemini 1.5 Pro	Google	-	1M+	Yes (Excellent multilingual)	2024-04	Closed	Massive context window, efficient
Gemini 1.5 Flash	Google	-	1M+	Yes (Excellent multilingual)	2024-05	Closed	Faster inference, cost-effective
Gemini 2.0 Flash	Google	-	1M+	Yes (Excellent multilingual)	2024-12	Closed	Latest generation, improved speed
LLaMA 1	Meta	7/13/33/65B	2048	Yes (Good multilingual base)	2023-02	Open	Foundation for open-source LLMs
LLaMA 2	Meta	7/13/70B	4096	Yes (Good multilingual base)	2023-07	Open	Commercial license, improved
LLaMA 3	Meta	8/70B	128K	Yes (Improved multilingual)	2024-04	Open	Enhanced capabilities, longer context
LLaMA 3.1	Meta	8/70/405B	128K	Yes (Improved multilingual)	2024-07	Open	Multi-modal, larger scale
LLaMA 3.3-70B	Meta	70B	128K	Yes (Improved multilingual)	2024-12	Open	Optimized 70B version, enhanced performance
Mistral 7B	Mistral AI	7.3B	8192	Yes (Strong in European languages)	2023-09	Open	High efficiency, strong performance
Mistral 8B	Mistral AI	8B	128K	Yes (Strong multilingual performance)	2024-10	Open	Larger capacity, long context support
Mistral 8x7B	Mistral AI	46.7B	32K	Yes (Strong multilingual performance)	2023-12	Open	Mixture of Experts, open weights
Mistral 8x22B	Mistral AI	141B	64K	Yes (Strong multilingual performance)	2024-04	Open	Larger MoE, improved performance
Mistral Large	Mistral AI	-	32K	Yes (Strong multilingual performance)	2024-02	Closed	Strong reasoning capabilities
Gemma 2B	Google	2.5B	8192	Yes (Designed for multilingual use)	2024-02	Open	Lightweight, Google's open model
Gemma 7B	Google	7B	8192	Yes (Designed for multilingual use)	2024-02	Open	Balanced performance and efficiency
Gemma 2	Google	9/27B	8192	Yes (Enhanced multilingual)	2024-06	Open	Improved architecture, better performance
Gemma 3-12B	Google	12/27B	8192	Yes (Enhanced multilingual)	2024-06	Open	Enhanced instruction-following
Claude 1	Anthropic	-	9000	Limited (Primarily English-optimized)	2023-03	Closed	Safety-focused, constitutional AI
Claude 2	Anthropic	-	100K	Limited (Primarily English-optimized)	2023-07	Closed	Extended context, improved coding
Claude 3	Anthropic	-	200K	Limited (Better but still English-focused)	2024-03	Closed	Multi-modal, state-of-the-art
Claude 3.5	Anthropic	-	200K	Limited (Improved but not fully multilingual)	2024-06	Closed	Enhanced reasoning, lower cost
Phi-1	Microsoft	1.3B	2048	Limited (Mainly English)	2023-06	Open	Small but capable, "textbooks" data
Phi-1.5	Microsoft	1.3B	2048	Limited (Mainly English)	2023-09	Open	Improved reasoning capabilities
Phi-2	Microsoft	2.7B	2048	Limited (English-focused)	2023-12	Open	State-of-the-art for small models
Phi-3	Microsoft	3.8/7/14B	128K	Limited (Better but still English-optimized)	2024-04	Open	Mobile-optimized, long context
Phi-4	Microsoft	14.7B	8192	Limited (English-focused)	2024-12	Open	Large-scale, instruction-tuned
Falcon 7B	TII	7B	2048	Yes (Good multilingual foundation)	2023-03	Open	Open-source, multilingual
Falcon 10B	TII	10.3B	32K	Yes (Good multilingual foundation)	2024-12	Open	High performance, reasoning, code
Falcon 40B	TII	40B	2048	Yes (Good multilingual foundation)	2023-03	Open	Large-scale open model
Falcon 180B	TII	180B	2048	Yes (Good multilingual foundation)	2023-09	Open	One of largest open models
DeepSeek-V2	DeepSeek AI	236B (active)	128K	Yes (Strong multilingual capabilities)	2024-05	Open	Efficient MoE architecture
DeepSeek-V3	DeepSeek AI	671B	128K	Yes (Strong multilingual capabilities)	2024-12	Open	Enhanced performance
DeepSeek-R1	DeepSeek AI	685B	128K	Yes (Strong multilingual capabilities)	2025-01	Open	Reasoning-optimized, specialized in complex tasks

Collectively, these families illustrate the diversification of the LLM ecosystem. While early models like GPT and LLaMA focused on scaling and performance, newer approaches increasingly emphasize efficiency, openness, interpretability, and alignment—marking an evolution from sheer size toward balanced, accessible, and ethically guided model design.

Table 2.1 provides a comprehensive overview of prominent LLMs discussed in this work, summarizing their technical specifications, multilingual capabilities, and release timeline. The table highlights key differences in architecture, parameter scale, context length, and licensing, illustrating how LLMs have evolved in terms of both performance and accessibility. Open-weight models such as LLaMA, Mistral, Gemma, Phi, Falcon, and DeepSeek demonstrate the growing trend toward transparency, efficient scaling, and multilingual support, whereas proprietary models like GPT, Gemini, and Claude emphasize commercial deployment, multimodal reasoning, and advanced instruction-tuning. By organizing the models chronologically and including key technical metrics, this table allows for a clear comparison of design priorities and capabilities, providing a valuable reference for understanding current trends and trade-offs in LLM development.

2.6 Summary of Word Embedding and LLM Evolution

The development of word embeddings and vector space models represents a progressive shift in how language meaning is represented and computed. As shown in Table 2.2, this evolution can be viewed as a continuum from early statistical representations to modern transformer-based architectures, reflecting both conceptual and technological advances in natural language processing.

The first generation of models, such as Bag-of-Words and TF-IDF, relied on sparse, count-based representations. These approaches were intuitive and interpretable but fundamentally limited by their inability to capture semantic relationships or contextual information. Words were treated as independent symbols, and meaning was inferred solely from frequency statistics, leading to high-dimensional yet semantically shallow representations.

The advent of neural-based embeddings, initiated by models like Word2Vec, GloVe, and FastText, marked a decisive breakthrough. These models introduced dense, distributed representations that encoded semantic regularities in continuous vector spaces. For the first time, it became possible to measure word similarity and perform vector arithmetic over meanings, capturing analogies such as *king* – *man* + *woman* $\approx$ *queen*. However, these embeddings remained *static*: a word was associated with a single vector, regardless of its context, limiting their expressiveness in handling polysemy or contextual nuances.

The introduction of contextual embeddings with ELMo and later BERT transformed this paradigm. Instead of assigning one fixed representation per word, these models generated embeddings that varied according to the surrounding linguistic context. BERT's transformer-based encoder architecture, leveraging the self-attention mechanism, enabled deep bidirectional understanding of text and became a foundational model for numerous downstream NLP tasks. This contextualization represented a major leap toward capturing the dynamic and relational nature of meaning in language.

The subsequent rise of large language models (LLMs) such as GPT, LLaMA, Mistral, and other contemporary families extended this approach even further. Built upon transformer decoder architectures, these models combined massive pre-training on diverse corpora with fine-tuning techniques that enabled advanced reasoning, multilingual capabilities, and text generation. Recent families such as Gemma, Claude, Falcon, Phi, and DeepSeek illustrate the growing diversity of LLM design philosophies—some emphasizing efficiency and open access, others focusing on safety, alignment, or multimodal reasoning.

Overall, the trajectory outlined in Table 2.2 highlights a clear conceptual and technological progression: from symbolic and frequency-based models, to distributed and contextual embeddings, and

finally to large-scale generative systems. Each generation has built upon the limitations of its predecessors, progressively bringing computational representations of language closer to capturing the complexity and richness of human semantics. This continuous evolution underpins many of today's advances in both academic research and applied domains such as finance, labour market intelligence, and beyond.

Table 2.2: Comparison of major word embedding and large language model (LLM) families. Models are listed in chronological order, illustrating the evolution from early statistical approaches to transformer-based architectures.

Model / Family	**Architecture**	**Representation**	**Main Features / Limitations**
TF-IDF	Count-based statistical	Static	Simple and interpretable; ignores word order and semantic similarity
Word2Vec	Neural Network	Static	Captures semantic relations; limited context sensitivity
GloVe	Matrix factorization	Static	Combines global and local context; context-independent
FastText	Neural Network	Static	Handles morphology and rare words; limited context modeling
ELMo	BiLSTM-based deep	Contextual	Context-dependent embeddings; computationally heavier
BERT	Transformer (encoder)	Contextual	Deep bidirectional context; high resource requirements
GPT family	Transformer (decoder)	Contextual	Strong generative capability; unidirectional context
LLaMA family	Transformer (decoder)	Contextual	Open-weight, efficient, multilingual
Mistral family	Transformer (decoder)	Contextual	Performance and efficiency; long context windows
Gemma family	Transformer (decoder)	Contextual	Lightweight, open, efficient
Claude family	Transformer (decoder)	Contextual	Safety, reasoning, conversational alignment
Phi family	Transformer (decoder)	Contextual	Small, high-performing, reasoning optimized
Falcon family	Transformer (decoder)	Contextual	Open-weight, multilingual, efficient
DeepSeek family	Transformer (decoder)	Contextual	Large-scale efficiency, multilingual, reasoning

Chapter 3

Financial Natural Language Processing

Natural Language Processing (NLP) in the financial domain has emerged as a pivotal tool for extracting insights from the vast and continuously growing body of textual data. Financial texts include news articles, earnings reports, regulatory filings, analyst notes, social media posts, and forum discussions, all of which contain valuable signals regarding market sentiment, company performance, and potential risks. The ability to systematically process and quantify such textual information has profound implications for asset pricing, risk management, trading strategies, and regulatory compliance.

Financial NLP leverages computational techniques to transform unstructured text into structured, machine-readable representations. By doing so, it enables the application of statistical and machine learning models for predictive and descriptive tasks. These representations can capture semantic relationships, contextual nuances, and patterns that may not be readily apparent through traditional quantitative indicators. Importantly, the integration of textual data with numerical financial data enhances the predictive power of models and supports a more holistic understanding of market dynamics.

Recent advancements in vector space models (VSMs), word embeddings, and large language models (LLMs) have significantly expanded the capabilities of Financial NLP. Dense embeddings allow for the quantification of semantic similarity, sentiment intensity, and event relevance, providing richer features than conventional bag-of-words approaches. Contextual embeddings and transformer-based models enable the disambiguation of polysemous terms and a deeper understanding of complex financial narratives, including forward-looking statements and nuanced market commentary.

From a practical perspective, Financial NLP is employed in several key applications: sentiment analysis for news and social media monitoring, predictive modeling of stock prices or volatility, event detection for mergers and acquisitions or corporate announcements, and the identification of correlations among financial instruments based on textual co-occurrences. These applications demonstrate how the combination of advanced NLP techniques with domain-specific financial knowledge can yield actionable insights for investors, analysts, and regulators.

A typical pipeline in Financial NLP is composed of different steps:

1. It begins with the collection of *raw financial texts*, which may include sources such as corporate filings, earnings reports, news articles, analyst commentaries, or social media posts. These sources are rich in information but unstructured, requiring several preprocessing steps before they can be analyzed computationally.

2. The *preprocessing phase* involves cleaning and normalizing the text, tokenizing sentences into words or subwords, and applying linguistic tools such as part-of-speech tagging, named entity recognition, or lemmatization. This step is crucial to ensure that subsequent models can accurately capture financial entities and relationships.

3. In the *text representation phase*, preprocessed text is transformed into numerical vectors through embedding models such as FatsText, BERT, or more LLMs. These embeddings encode seman-

tic and contextual information, allowing the system to recognize similarities between words, phrases, and entire documents within a financial context.

4. The resulting embeddings serve as inputs for the *financial modeling phase*, where predictive or descriptive algorithms are applied. Typical tasks include sentiment analysis, event detection, risk assessment, and forecasting of market movements. Advanced architectures, including transformer-based and graph neural networks, are increasingly adopted to model temporal dependencies and relationships among entities.

5. Finally, the system outputs structured information such as sentiment scores, trading signals, volatility forecasts, or risk indicators. These outputs can support a variety of financial applications—from algorithmic trading and portfolio optimization to regulatory monitoring and macroeconomic analysis.

This pipeline illustrates the progressive transformation of unstructured textual information into actionable, data-driven insights that complement traditional numerical analysis in finance.

3.1 Characteristics of Financial Language and Domain Challenges

Financial language presents a set of distinctive characteristics that make it fundamentally different from general-purpose text corpora on which most NLP models are trained. Unlike everyday language, financial text is dense, technical, and context-dependent, often combining formal reporting styles with subjective interpretations of market trends. This creates unique challenges for language modeling and semantic representation:

- One key aspect of financial language is its **high domain specificity**. Financial documents, such as earnings reports, analyst commentaries, and central bank statements, contain specialized terminology (e.g., *EBITDA*, *quantitative easing*, *yield curve inversion*) and implicit economic reasoning that general models fail to capture. Moreover, many terms have **polysemous meanings** depending on the context: for instance, *bullish*, *long*, or *short* may refer to sentiment, position, or strategy, depending on the text source.

- Another challenge comes from the **limited availability and proprietary nature of financial data**. While general NLP systems benefit from large and diverse corpora, financial texts are often restricted, specialized, and time-sensitive. This makes it challenging to train domain-specific embeddings without overfitting or introducing bias from datasets covering only specific periods or sources.

- Financial language also exhibits strong **temporal and event dependence**. The meaning and relevance of terms evolve with market conditions and economic cycles. Models trained on static corpora struggle to capture this dynamic semantic drift, emphasizing the need for **temporal adaptation** and **continual learning** in financial NLP.

- A further complexity comes from the **multimodal and heterogeneous nature** of financial information. Textual data interacts closely with numerical indicators, graphs, and structured data such as stock prices or macroeconomic variables. Understanding financial documents thus often requires reasoning over multiple modalities, blending linguistic and quantitative representations.

- Finally, **sentiment and pragmatics** play a central role in financial communication. Subtle linguistic cues such as hedging, tone, or framing can strongly influence market interpretations. Capturing these nuances requires embeddings capable of modeling fine-grained contextual meaning and rhetorical intent, going beyond surface-level word co-occurrence.

Together, these characteristics make financial NLP one of the most complex and high-stakes domains for vector space modeling. They motivate the development of specialized architectures and embeddings explicitly designed to encode financial semantics, adapt to temporal dynamics, and integrate multimodal signals.

3.2 Financial Word Embeddings and Domain Adaptation

While general-purpose word embeddings and language models have demonstrated remarkable capabilities in natural language understanding, they often fall short when applied directly to financial text. The domain-specific terminology, polysemy, temporal dependencies, and multimodal nature of financial language require representations that capture the nuanced and dynamic characteristics of this context. Models trained on generic corpora may fail to recognize subtle distinctions, interpret specialized terms correctly, or adapt to rapidly evolving market events.

To address these limitations, several strategies have been proposed for domain adaptation in financial NLP. One approach is **fine-tuning** pre-trained embeddings or language models on financial corpora, allowing them to retain general linguistic knowledge while learning domain-specific patterns. Another approach involves **training embeddings from scratch** on financial datasets, such as earnings reports, analyst notes, regulatory filings, or financial news articles. Hybrid methods have also been explored, combining general-purpose embeddings with financial lexicons or co-occurrence statistics to enhance semantic relevance and interpretability.

The introduction of transformer-based models marked a turning point in financial NLP. Among the first dedicated models, **FinBERT** represents a fine-tuned variant of BERT on large-scale financial corpora, including analyst reports and financial news. FinBERT has shown strong performance in sentiment analysis and text classification, benefiting from its contextual understanding of financial terminology and idioms.

More recently, the emergence of large language models (LLMs) trained directly on financial data has expanded the scope of financial NLP. **BloombergGPT**, developed by Bloomberg, constitutes one of the most significant efforts in this direction. It is a 50-billion-parameter model trained on a mixture of public financial documents and proprietary Bloomberg data, enabling both general-purpose reasoning and specialized financial analysis. Similarly, **FinGPT** proposes an open-source alternative built on a modular and transparent architecture, aiming to democratize access to finance-specific large language models. FinGPT combines public financial text data, reinforcement learning, and continual fine-tuning strategies to adapt rapidly to new financial information.

Together, these models illustrate a broader trend: the evolution from static word embeddings toward contextual and generative architectures capable of understanding, reasoning, and generating domain-specific financial knowledge. While the availability of high-quality financial data remains a limiting factor, such models mark an essential step toward bridging the gap between natural language understanding and real-world financial decision-making.

Extracting informative signals from financial documents often involves a combination of **contextual representation learning** and **linguistic feature engineering**. Many approaches rely on tone and sentiment analysis, keyword extraction, and contextual embeddings derived from transformer-based models such as FinBERT, which encode financial text into dense semantic vectors. These representations can be fused with other modalities—such as numerical or temporal data—through attention mechanisms or vector concatenation, enabling richer multimodal inference. Alternatively, linguistic feature engineering can be applied to extract interpretable information from text, such as domain-specific keywords or topic distributions using techniques like Latent Dirichlet Allocation (LDA) [29]. These features offer compact, high-level summaries of the underlying text and can serve either as direct input to the model or as a form of weak supervision [15].

Furthermore, sentiment analysis and emotion detection can provide additional signals by captur-

ing subjective aspects of the text. These can be obtained using pre-trained models like FinBERT or lexicon-based approaches such as those proposed by Loughran and McDonald [30], yielding sentiment polarity scores or emotion labels that enhance the model's ability to interpret market mood.

Overall, textual information can be represented through dense embeddings, symbolic features, topic-based summaries, or affective scores, depending on the nature of the task and its interaction with other data modalities.

3.3 Financial Textual Data Sources

The effectiveness of domain-specific embeddings and language models in finance depends critically on the choice and quality of the underlying textual corpora. Common sources can be categorized by source and intent.A first category includes **institutional and journalistic news**, such as those published by *Bloomberg*[1], *Reuters*[2], the *Financial Times*[3], *Yahoo Finance*[4], and *MarketWatch*[5]. These articles typically report on earnings results, mergers and acquisitions, regulatory changes, and geopolitical developments. Both headlines and full texts can be processed using natural language processing (NLP) techniques to estimate their relevance to financial markets or to extract sentiment polarity scores. A second type of textual source is represented by **regulatory and official documents**, such as statements from central banks (e.g., SEC 10-K and 10-Q reports, and FOMC[6] announcements) or formal corporate disclosures. These texts tend to be highly structured and formalized, yet they carry valuable information for financial analysis. A third relevant category includes **social media platforms**, such as X[7], reddit[8], and stocktwits[9]. These platforms provide real-time access to retail investor sentiment, rumors, and emergent narratives. Despite being noisy and informal, they have shown predictive value, especially for assets driven by retail trading activity. The integration of such domain-focused data with advanced modeling architectures enables the extraction of subtle semantic nuances, temporal dependencies, and sentiment cues that general-purpose embeddings often fail to capture.

Textual data used in financial tasks are typically obtained from either proprietary platforms or publicly available datasets. These sources provide high-quality, time-sensitive, and domain-relevant content that plays a key role in capturing market sentiment and identifying event-driven signals. In some cases, access to such corpora is granted through institutional collaborations or previously compiled datasets. For instance, the dataset introduced by Duan et al. [31] aggregates Reuters and Bloomberg articles published between October 2006 and December 2015, offering a rich resource for event-based financial modeling.

To promote reproducibility and cross-study comparability, several publicly available datasets have also been introduced. The *CMIN-US* dataset [32] combines large-scale financial texts with stock price time series from both U.S. and Chinese markets, enabling cross-market analyses of textual and numerical signals. Similarly, the widely used *ACL18* dataset [33] integrates two modalities—social media data and historical price information—comprising 387,045 tweets related to 70 companies across seven industries. Another valuable resource is the *2017 Earnings Conference Calls dataset* [34], which contains transcripts of corporate earnings calls collected from *Seeking Alpha*[10]. This dataset includes

[1] https://www.bloomberg.com
[2] https://www.reuters.com/
[3] https://www.ft.com/
[4] https://finance.yahoo.com/
[5] https://www.marketwatch.com/
[6] https://www.federalreserve.gov/monetarypolicy/fomc.htm
[7] https://x.com/
[8] https://www.reddit.com/
[9] https://stocktwits.com/
[10] https://seekingalpha.com/

detailed metadata such as speaker identities and corresponding utterances, facilitating fine-grained analyses of managerial tone, sentiment, and communication patterns.

Overall, the diversity and structure of financial textual data—from formal regulatory documents to informal social media content—provide complementary perspectives on market behavior. Selecting and combining these heterogeneous sources is thus a key step in developing robust and generalizable financial language models.

3.4 Financial NLP Tasks

In financial language processing, tasks can be broadly categorized into two complementary domains: *text understanding* and *predictive modelling*. The former focuses on extracting semantic, syntactic, or event-driven signals from financial texts, while the latter leverages such information—often in conjunction with numerical or temporal data—to predict market variables or guide investment decisions. The interplay between these two levels of analysis enables modern financial NLP systems to move from mere text interpretation to actionable financial insight.

Sentiment and Tone Analysis. Sentiment analysis has long been a cornerstone of financial text mining. Its goal is to assess the polarity or tone of financial discourse, identifying whether the expressed attitude toward a firm, market, or event is positive, negative, or neutral. Traditional approaches rely on domain-specific lexicons such as the *Loughran-McDonald* dictionary [30], while modern systems employ pretrained language models fine-tuned on financial corpora, such as *FinBERT* [35]. These models are particularly effective in handling context-dependent expressions (e.g., "beat estimates" or "underperform guidance") that standard sentiment models often misinterpret. Sentiment-derived indicators are frequently used as explanatory variables for market movements, volatility, and risk perception.

Event and Entity Extraction. Beyond sentiment, financial texts encode complex event structures involving entities such as firms, sectors, and regulatory bodies. Event extraction aims to identify and categorize such occurrences—e.g., earnings announcements, mergers and acquisitions, or policy changes—along with their participants and temporal attributes. Early works used rule-based or sequence labeling methods (e.g., BiLSTM-CRF), while recent studies increasingly leverage transformer-based encoders that can model long-range dependencies and domain-specific context [36, 37]. Event-level representations are also essential for causal inference and cross-document reasoning.

Causal and Temporal Reasoning. Recent advances have introduced tasks focused on uncovering causal or temporal dependencies between textual information and market outcomes. For instance, the *CMIN-US* dataset [32] explicitly links financial news to stock price reactions in the U.S. and Chinese markets, enabling models to learn directional cause–effect relations rather than simple correlations. These tasks mark a shift toward explainable and reasoning-based financial NLP.

Predictive Modelling Tasks. Beyond textual understanding, word and language embeddings play a crucial role in predictive financial modelling, where textual representations are integrated with numerical and temporal features to forecast market behavior. In this context, predictive tasks leverage embeddings to transform unstructured financial texts—such as news, reports, or social media messages—into quantitative signals that inform decision-making models.

A first class of problems, **stock feature prediction**, aims to estimate specific quantitative indicators for individual assets. Among these, *stock movement prediction* is one of the most widely studied and is typically cast as a classification task that determines whether a stock price will rise, fall, or

remain stable:

$$y_t = \begin{cases} 1 & \text{if } P_{t+1} > P_t \\ 0 & \text{otherwise} \end{cases}. \tag{3.1}$$

Other variants include *stock return prediction*, where models forecast the future return $r_{t+\Delta t} = \frac{P_{t+\Delta t} - P_t}{P_t}$, and tasks such as volatility or index prediction, which extend the same logic to aggregated or risk-related measures.

A second family of problems, **stock ranking**, focuses instead on the relative ordering of assets according to predicted performance. The model assigns a score $r_i = f_\theta(s_i)$ to each stock, producing a ranked list $\{s_{(1)}, s_{(2)}, \ldots, s_{(n)}\}$ satisfying $r_{(1)} \geq r_{(2)} \geq \cdots \geq r_{(n)}$. This formulation is particularly effective for portfolio optimization and long–short strategies [38], as it prioritizes comparative relationships over absolute forecasts.

Overall, these predictive modelling tasks exemplify how textual embeddings—ranging from static vectors to contextual representations—can be transformed into actionable financial signals, bridging the gap between natural language understanding and quantitative forecasting.

3.5 Summary of Word Embeddings and Language Models for Finance

In summary, the evolution of word embeddings and language models has profoundly influenced financial natural language processing. From early static representations such as Word2Vec, GloVe, and FastText to contextualized and generative models like BERT, FinBERT, and BloombergGPT, the field has progressively shifted toward architectures capable of understanding domain-specific semantics and reasoning over complex textual data. The introduction of finance-oriented corpora—including regulatory filings, news articles, social media posts, and earnings call transcripts—has enabled the construction of embeddings that capture temporal and contextual dependencies unique to financial language.

These representations have been applied across a wide spectrum of predictive modelling tasks, ranging from sentiment and event detection to stock movement forecasting and cross-market analysis. The ability to extract latent signals from unstructured financial text has proven critical for understanding market dynamics, assessing investor sentiment, and supporting decision-making processes.

Chapter 4

Applications in Labour Market Intelligence (LMI)

In recent years, the role of specialized web portals and services in intermediation has grown exponentially. This surge has given rise to the concept of Labor Market Intelligence (LMI), which involves leveraging AI algorithms and frameworks to analyze labor market data and support data-driven decision-making (see, e.g., [39, 40, 41, 42, 4, 5, 43]). In this context, the ability to monitor, analyze, and understand labor market changes both (i) in a timely manner and (ii) at a highly granular geographical level has become increasingly significant. Online Job Advertisements (OJAs) possess these features and have become increasingly important for academic research and the development of innovative statistics in the last years. Academic research has exploited the granularity of OJAs to analyze labor market concentration [44, 45, 46, 47], but most importantly has leveraged the information contained in the text of the advertisement to analyze firms' skill requirements [48, 49, 50, 51, 10].

Figure 4.1 shows how the labour market is strongly influenced by various factors: the political, social and technological context in which one lives, education, whether at school or from other contexts, together with experience define what are the soft and hard skills of an individual. These aspects affect the evolution of the labour market and how it is analyzed through LMI tools, the changes in the labour market also affect the aspects mentioned above, forming a cycle of continuous transformations.

This is relevant not only for research purposes but also for the production of statistical data supporting skill-related policies. In the European context in 2016 the European Commission issued the communication "A New Skills Agenda for Europe"[1] highlighting a number of actions and initiatives aimed at equipping the European labor force with the skills of the future. In support of this initiative, the EU agency Cedefop subsequently teamed up with Eurostat to develop a system capable of collecting and classifying online job vacancies for the entire EU, covering all 28 Member States and the Union's 32 languages [53]. The result of this effort is the Web Intelligence Hub (WIH), which has collected OJAs from more than 1000 sources in Europe since 2019. Several results from this effort have been published in studies such as [8, 9, 10, 11, 12]. This study leverages this initiative by utilizing the knowledge base compiled by the WIH to train and optimize word embeddings.

The importance of LMI has grown a lot in recent years, also driven by the rapid technological evolution that produces more and more tools that can also be used in this scenario, such as all the techniques for extracting and processing information from texts written in natural language, including online job advertisements (OJAs), curriculum vitae (CV), and online professional profiles.

Despite their potential, LMI applications face several challenges. The data are often noisy, domain-specific, and vary significantly in terminology and linguistic structure across sectors, regions, and languages. Furthermore, labour market texts are characterized by high lexical variability and rapid

[1] COM(2016) 381/2, available at https://migrant-integration.ec.europa.eu/sites/default/files/2020-07/SkillsAgenda.pdf

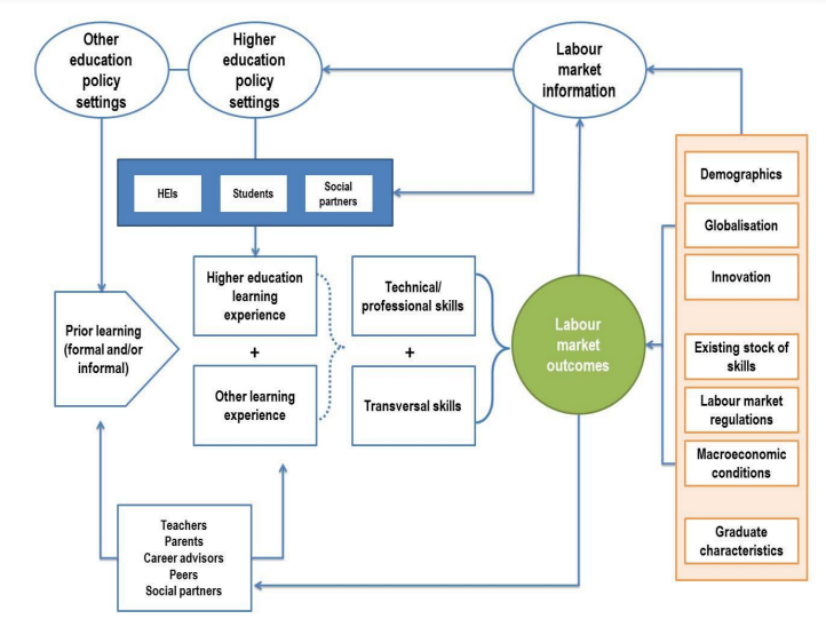


Figure 4.1: Analytical framework for the in-depth analysis of the labour market relevance and outcomes of higher education systems[52].

temporal evolution, reflecting emerging technologies and occupational shifts. Embedding-based representations help address these challenges by mapping words, skills, and occupations into continuous vector spaces, where semantic similarity corresponds to geometric proximity. This representation facilitates the measurement of conceptual similarity, supports clustering and matching tasks, and provides a foundation for multilingual and cross-country analysis.

The interest in this sector involves not only private companies, but also and increasingly, public organizations and national governments that are investing heavily in the creation of tools with which to analyze the labor market. An example of this are the on-line portals of Canada or some American states; the European Union is also heavily involved in this sector, an example is the ESCO taxonomy for the cataloging of occupations and skills present in the European labor market or EURES, a European cooperation network of employment services that has the purpose to facilitate the free movement of workers and to provide equal opportunities for all European citizens.

4.1 Labour Market Intelligence Data Sources

The effectiveness of LMI systems depends critically on the quality and diversity of their data sources. Modern pipelines typically integrate both unstructured textual data and structured knowledge bases to capture complementary aspects of labour market dynamics. Textual sources, such as job advertisements and curricula vitae, provide high-frequency, real-world evidence of skill demand and supply, while structured taxonomies and ontologies, such as ESCO, offer standardized frameworks for classification, interoperability, and cross-country comparison.

Data for LMI applications are collected from a variety of channels. The most common are online job portals (e.g., *LinkedIn*, *Indeed*, *Monster*, *EURES*), national public employment services, and corporate career pages. These platforms offer large-scale, continuously updated streams of job postings that can be scraped, parsed, and analyzed. Complementary sources include CV repositories and professional networking platforms, which provide insights into individual skills and career paths.

4.1.1 Unstructured Textual Sources

Among the most valuable unstructured sources for LMI are **job advertisements (OJAs)** and **curricula vitae (CVs)**. These documents represent two complementary perspectives on the labour market: OJAs reflect the demand for skills, while CVs describe the supply side in terms of individual competencies and experiences.

Job advertisements are semi-structured texts typically containing information on job titles, required skills, education levels, professional experience, and sometimes salary ranges or contract types. They provide real-time, fine-grained indicators of occupational demand and emerging trends in the labour market. Because they are written in natural language, OJAs exhibit substantial linguistic variability, including synonymy (e.g., "data scientist" vs. "machine learning engineer"), ambiguity, and cross-country or sectoral differences in terminology. Natural language processing and word embedding models are therefore essential to extract relevant information, normalize job titles, and link job postings to standardized occupational categories.

Curricula vitae and professional profiles, on the other hand, offer insights into the skills and experiences of the labour force. These documents often vary widely in format and level of detail, but they contain structured elements such as education history, previous employment, and self-declared skills. Aggregated CV data can be used to estimate skill availability, identify emerging competencies, or detect mismatches between skill supply and demand. However, privacy constraints and access limitations mean that CV-based analyses are typically restricted to anonymized or aggregated datasets. When available, their integration with OJAs provides a comprehensive view of labour market imbalances and transitions.

4.1.2 Occupational Taxonomies

A *taxonomy* is a structured classification system that organizes concepts into a hierarchical framework based on their relationships and levels of generality [54, 55, 56, 6]. According to [57], a taxonomy can be defined as:

Definition 4.1.1 (Taxonomy)**.** A *taxonomy* is defined as a couple $T = (C, H_C)$, where:

- C is the set of concepts $c \in C$ belonging to the domain of interest (i.e., the nodes of the taxonomy).
- H_C is a directed taxonomic binary relation between concepts, such that $H_C \subseteq \{(c_i, c_j) \mid (c_i, c_j) \in C^2, i \neq j\}$. This relation, denoted as $H_C(c_1, c_2)$, indicates that c_1 is a sub-concept of c_2, also known as the *IS-A* relation.

Taxonomies play a crucial role in various domains where they are used to structure knowledge in ontologies and controlled vocabularies. In Natural Language Processing (NLP) and labor market analysis, taxonomies such as ESCO[2] (European Skills, Competences, and Occupations) provide a structured representation of skills and occupations, facilitating semantic interoperability across languages and regions. ESCO can be defined as follows:

Definition 4.1.2 (ESCO Taxonomy)**.** *ESCO works as a dictionary, describing, identifying and classifying professional occupations and skills relevant for the EU labour market and education and training. Those concepts and the relationships between them can be understood by electronic systems, which allows different online platforms to use ESCO for services like matching jobseekers to jobs on the basis of their skills, suggesting trainings to people who want to reskill or upskill.*

ESCO is translated into 27 languages (all official EU languages plus Icelandic, Norwegian and Arabic) and describes 2942 occupations and 13,890 skills required for these occupations. ESCO

[2] `https://esco.ec.europa.eu/en/classification`

defines a common foundation on jobs and skills and supports labour mobility across the European community and a more integrated and efficient labor market. ESCO can be used by all those actors interested in issues related to employment, education and training.

The first version of ESCO was published on 28 July 2017 and it is available on an online portal and can be consulted free of charge. It is constantly updated to better represent all the news from the world of work. ESCO is organized into three pillars, which are strongly connected to each other as shown in figure 4.2:

- The occupations pillar: the pillar of occupations has a hierarchical structure, descending the hierarchy you find more specific professions than their fathers.

- The knowledge, skills and competences pillar: knowledge is defined as facts, principles, theories and practices to which it is linked a field of work or study. Skills mean the ability to complete tasks and solve problems apply knowledge and competence is the proven ability to use personal, social and methodological skills.

- The qualifications pillar: The qualifications pillar aims to collect existing information on qualifications come from national qualifications databases of Member States.

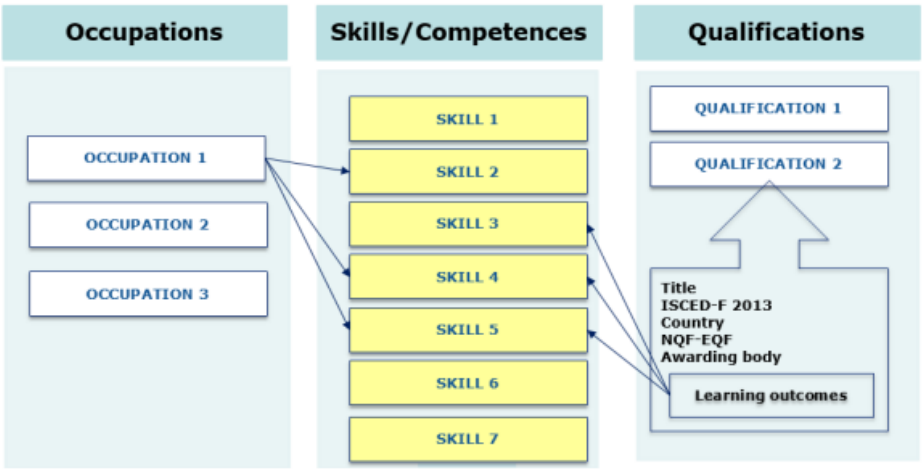


Figure 4.2: Linking qualifications with the occupations and skills pillars[58].

In a taxonomic structure, concepts are typically arranged in a parent-child relationship, where broader categories encompass more specific subcategories. For example, in ESCO, the occupation *2512 - Software Developer* is a sub-concept of the broader concept *251 - Software and applications developers and analysts*, which in turn belongs to the more general concept of *2 - Professionals*. These hierarchical relationships can be leveraged to compute concept similarity, which is fundamental for tasks such as job matching, skills inference, and occupational classification.

Several measures are available to evaluate the similarity between concepts in a taxonomy. These measures can be broadly classified into two main categories: *path-based* and *Information Content (IC)-based* approaches. Path-based measures estimate similarity by analyzing the structure of the taxonomy, typically using the shortest path between two concepts or considering the depth of their lowest common ancestor, as proposed in [59]. IC-based measures define similarity based on the probability of encountering a concept, assuming that more specific concepts carry higher information content. Approaches of this kind have been presented by [60, 61, 62].

4.1.3 The Hierarchical Semantic Similarity at a glance

For our studies, we use the Hierarchical Semantic Similarity (HSS) developed by [55] as a similarity measure. The measure has then been implemented as a tool to perform taxonomy refinement [63]. The HSS has been designed to measure the degree to which two concepts are related within a given taxonomy.

Intuitively, the metric is based on the concept of information content, which states that the lower the rank of a concept $c \in \mathcal{C}$ that contains two entities, the higher the information content (*IC*) the two entities share. According to information theory, the *IC* of a concept c can be approximated by its negative log-likelihood: $IC(c) = -\log p(c)$ where $p(c)$ is the probability of encountering the concept c. Following [64], one can supplement the taxonomy with a probability measure $p : \mathcal{C} \to [0,1]$ such that for every concept $c \in \mathcal{C}$, $p(c)$ is the probability of encountering an instance of the concept c. It follows that p (i) is monotonic and (ii) decreases with the rank of the taxonomy, i.e., if c_1 is a sub-concept of c_2, then $p(c_1) \leq p(c_2)$. This means the probability decreases as we move to more specific (deeper) concepts in the taxonomy. To estimate the values of p, [64] employs the frequency of concepts within a large text corpus. However, we aim to infer the similarity values inherent to the semantic hierarchy to extend a taxonomy constructed by human experts. In this context, the Hierarchical Semantic Similarity measure is particularly useful, as it leverages the frequencies of concepts and entities within the taxonomy to compute the value of p. Specifically, it estimates p as: $\hat{p}(c) = \frac{N_c}{N}$ where N is the cardinality, i.e., the number of entities (words), of the taxonomy and N_c the sum of the cardinality of the concept c with the cardinality of all its hyponyms.

Note that $\hat{p}(c)$ is monotonic and increases with the size and generality of the concept c (i.e., N_c is larger for more general concepts), thus correctly reflecting the defined properties of p.

Given two words w_1 and w_2, [64] defines $c_1 \in s(w_1)$ and $c_2 \in s(w_2)$ all the concepts containing w_1 and w_2 respectively, i.e. the *senses* of w_1 and w_2. Therefore, there are $S_{w_1} \times S_{w_2}$ possible combinations of their word senses, where S_{w_1} and S_{w_2} are the cardinality of $s(w_1)$ and $s(w_2)$ respectively. Given that, [55] define $\mathcal{L}$ as the set of all the lowest common ancestor (LCA) for all the combinations of $c_1 \in s(w_1), c_2 \in s(w_2)$. The hierarchical semantic similarity between the words w_1 and w_2 can be defined as:

Definition 4.1.3 (Hierarchical Semantic Similarity (HSS))**.** The semantic similarity between two words w_1 and w_2 is computed as:

$$\texttt{HSS}(w_1, w_2) = \sum_{\ell \in \mathcal{L}} \hat{p}\,(\ell = L \mid w_1, w_2) \times IC(L) \tag{4.1}$$

where $\hat{p}\,(\ell = L \mid w_1, w_2)$ is the probability of ℓ being the Least Common Ancestor (LCA) of w_1 and w_2, computed using Bayes' theorem as:

$$\hat{p}\,(\ell = L \mid w_1, w_2) = \frac{\hat{p}\,(w_1, w_2 \mid \ell = L)\,\hat{p}\,(L)}{\hat{p}\,(w_1, w_2)} \tag{4.2}$$

Then, we define N_ℓ as the cardinality of ℓ and all its descendants. The numerator can be rewritten as:

$$\hat{p}\,(w_1, w_2 \mid \ell = L)\,\hat{p}\,(L) = \frac{S_{<w_1, w_2> \in \ell}}{|descend(\ell)|^2} \times \frac{N_\ell}{N} \tag{4.3}$$

where the first leg of the *rhs* is the class conditional probability of the pair $< w_1, w_2 >$ and the second one is the marginal probability of class ℓ. The term $|descend(\ell)|$ represents the number of subconcepts of ℓ. Since they could have at most one word sense w_i for each concept c, $|descend(\ell)|^2$ represents the maximum number of combinations of word senses $< w_1, w_2 >$ which have ℓ as LCA. $S_{<w_1, w_2> \in L}$ is the number of pairs of senses of word w_1 and w_2 which have L as LCA and $\frac{S_{<w_1, w_2> \in \ell}}{|\text{descend}(\ell)|^2}$ is the proportion of this maximum that is actually realized by the senses of w_1 and w_2. The denominator can be written as:

$$\hat{p}(w_1, w_2) = \sum_{k \in \mathcal{L}} \frac{S_{<w_1, w_2> \in k}}{|descend(k)|^2} \tag{4.4}$$

Benefits of using HSS. Unlike traditional measures that rely solely on lexical similarity or corpus-based co-occurrence, HSS explicitly considers the taxonomic distance between concepts, taking into account both their hierarchical depth and common ancestors. Given two occupations in a taxonomy, their HSS score reflects the degree to which they share commonalities: occupations that are direct siblings (i.e., sharing the same parent node) or have a close ancestor will exhibit a higher similarity than those that belong to distant branches of the hierarchy. By leveraging HSS, we can assess how well a word embedding model preserves the relationships defined in a taxonomy, providing an intrinsic evaluation of its effectiveness in capturing domain-specific knowledge. The following sections describe how we utilize this metric to compare different embedding models and identify those that best preserve the ESCO taxonomy relationships.

Structured taxonomies and ontologies play a central role in interpreting and organizing information extracted from unstructured sources. They provide a shared conceptual framework for describing occupations, skills, and qualifications, enabling semantic interoperability across datasets, systems, and countries. Integrating unstructured data with these structured frameworks is a key challenge in LMI research. Vector-based semantic models, such as word or sentence embeddings, are increasingly used to align textual content (e.g., job descriptions or extracted skills) with taxonomy concepts. This alignment enables tasks such as occupation classification, skill standardization, and cross-country skill matching. By bridging the gap between unstructured text and structured ontologies, such methods enhance the interpretability and analytical depth of modern Labour Market Intelligence systems.

4.2 Vector Space Model & Labour Market Intelligence

Word embeddings and contextual language models have become central to LMI systems; these methods provide dense vector representations of textual elements such as occupations, skills, and qualifications, capturing latent relationships that traditional symbolic or keyword-based approaches fail to detect. By mapping job-related terms into a continuous semantic space, embeddings allow for similarity-based reasoning, clustering, and alignment across countries and languages.

In the context of LMI, unstructured textual sources such as OJAs and CVs are particularly challenging due to their informal style, domain-specific terminology, and multilingual nature. Word embeddings trained on large collections of OJAs can model the semantic proximity between occupational titles and skill expressions, allowing systems to infer relationships that are not explicitly stated in the text. For instance, embeddings can capture that *data scientist* is semantically close to *machine learning engineer*, or that *Python* and *TensorFlow* often co-occur in similar professional contexts. This ability is crucial for skill matching, job recommendation, and occupational analysis.

Beyond static representations such as FastText, which remain valuable for their efficiency and robustness to rare or unseen terms, transformer-based models have significantly advanced NLP applications in the financial and labour market domains. Adaptations of the BERT architecture fine-tuned for tasks such as skill extraction and occupation classificationnto employment-related text enable the learning of contextual embeddings that capture the semantics of occupations and skills.

These embedding-based frameworks thus provide the foundation for linking textual and structured data within LMI ecosystems. They enable tasks such as automatic skill tagging, occupation–skill matching, labour market trend detection, and cross-country comparison. In this way, representation learning acts as a bridge between raw textual evidence and high-level policy insights, supporting a data-driven understanding of workforce evolution and skill transformation in an increasingly multilingual European labour market.

Motivating Example of Word Embeddings for Labor Market

A graphical representation of a word embedding model is shown in Fig. 4.3, trained on millions of online job advertisement titles. The map highlights existing concepts from the ESCO taxonomy (represented by empty shapes) and alternative labels (terms that emerge from online data and are strongly related to ESCO concepts but are not yet included in ESCO).
The embedding model effectively *"encodes"* words with similar meanings within the context of the labor market. For example, while a *data engineer* and a *data scientist* are both categorized as sub-concepts under the *2511: System Analyst* ISCO group in ESCO, their real-world roles differ significantly—something any computer scientist would recognize. In practice, a data engineer often aligns more closely with *2521: Database Designers and Administrators* than with its theoretical ISCO group. Conversely, there are instances where the taxonomy accurately reflects real labor market demands. A clear example is *3521: Broadcasting and Audio-Visual Technicians*, which forms a tight cluster on the map, indicating a strong alignment between de-facto and de-jure labor market occupations. Similarly, *3513: Computer Network and Systems Technicians* also demonstrate this consistency, albeit to a slightly lesser extent. Notably, representing words as semantic vectors in vector space enables the use of algebra to perform operations over concepts. This means one can perform the following vector operation:
$\vec{v}_{it\_security\_manager} - \vec{v}_{security} + \vec{v}_{data} = \vec{v}_{data\_quality\_manager}$
This highlights the capability to reason mathematically over words and their semantic relationships using vector algebra. Consequently, it underscores the importance of generating high-quality embeddings to ensure the inference process remains accurate and free from unintended biases. Poorly constructed embeddings can distort semantic relationships, leading to misleading conclusions and reinforcing existing biases within the data.
On the other side, the UMAP plot in Fig.4.3 illustrates the embedding that best aligns with the ESCO taxonomy, as evaluated by the HSS metric introduced by [55]. To emphasize the effect of using a non-optimized word embedding model—trained with default parameters—Fig. 4.4 presents an alternative embedding generated from the same dataset, without validation against the ESCO taxonomy or the application of grid search optimization.

Natural Language Processing and representation learning techniques have enabled a wide range of applications in Labour Market Intelligence (LMI), supporting both micro-level analyses (e.g. skill extraction from job descriptions) and macro-level insights (e.g. monitoring skill trends across countries). By leveraging word and sentence embeddings trained on large textual corpora such as job advertisements and curricula, these systems are able to capture semantic relationships between occupations, skills, and qualifications, bridging unstructured text and structured taxonomies such as ESCO.

Skill Extraction and Normalisation A fundamental task in LMI is the automated identification and normalisation of skill mentions from unstructured text (e.g., job postings, CVs) to a standardised taxonomy. The process typically follows a two-stage pipeline: (1) *Named Entity Recognition (NER)* for detecting skill phrases, and (2) *Skill Linking* for mapping these phrases to canonical entities within a reference ontology. The field has been advanced by deep contextual language models, which leverage contextualised embeddings to disambiguate complex, multi-word, or overlapping skill expressions with high precision.

Occupational Classification Occupational classification involves assigning standardised codes to job descriptions or CVs, a process vital for ensuring data comparability, statistical analysis, and applications like job matching. Early approaches relied on traditional classifiers using bag-of-words

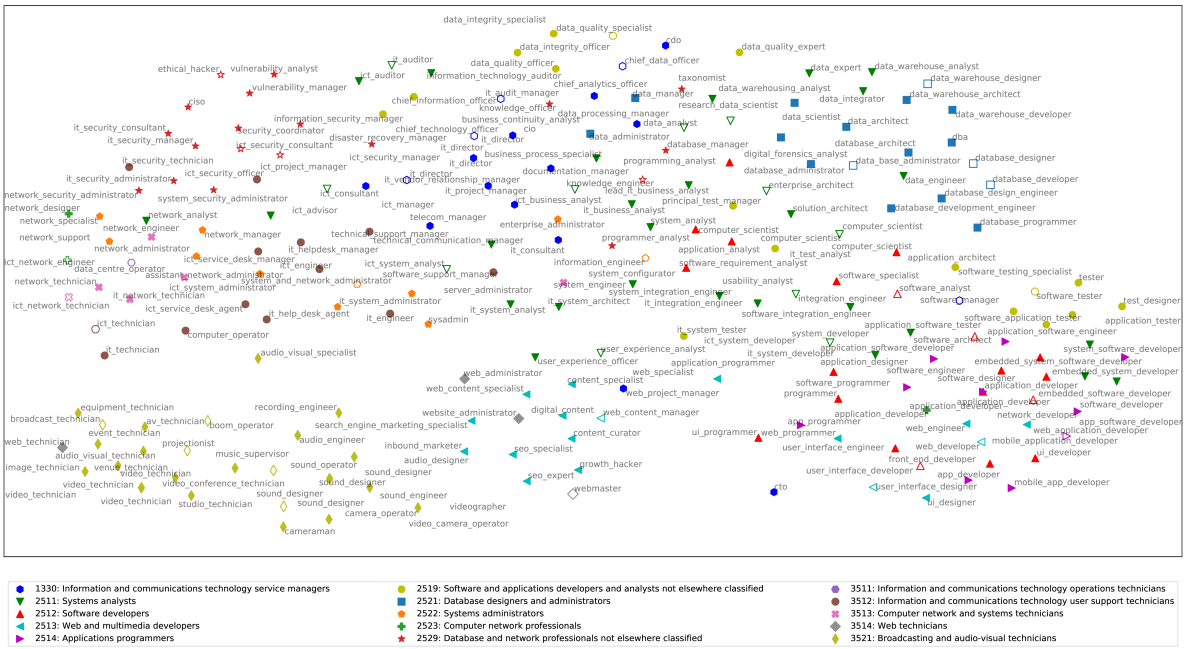


Figure 4.3: UMAP plot of the **best** word embedding model, according to HSS metric [55]. The plot illustrates the ESCO concepts, and words belonging to each group are displayed, distinguishing between narrower occupations (empty shapes) and alternative labels (filled shapes). Trained over 2 million ICT-related jobs in the UK. Taken from [55]

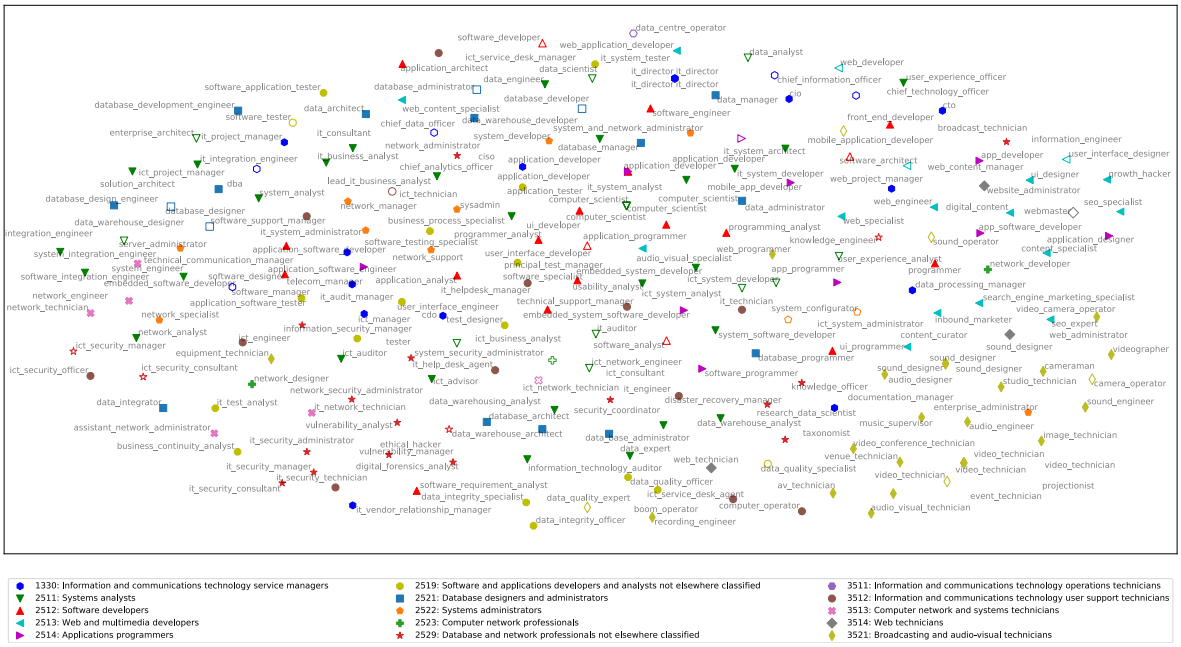


Figure 4.4: UMAP plot of the **worst** word embedding model, according to HSS metric [55]. The plot shows the ESCO concepts and words belonging to each group are shown distinguishing between narrower occupations (shallow shape) and alternative labels (filled shape). Trained over 2 million ICT-related jobs in UK

representations, which required significant feature engineering. Contemporary methods utilise neural architectures built on contextual embeddings, which better capture semantic nuances and facilitate cross-lingual classification. This allows for consistent categorisation across different labour markets, even for lower-resource languages.

Job–Skill Matching and Recommendation A direct application of these techniques is job–skill matching, which aims to align job vacancies with candidate profiles or skill sets. This task primarily relies on computing semantic similarity within a shared embedding space, often after a normalisation step. Methodologies range from graph-based models, framing the problem as link prediction, to modern zero-shot approaches leveraging Large Language Models (LLMs). These latter systems can generate recommendations based on semantic similarity and ranking, potentially bypassing the need for extensive labelled training data.

4.3 Summary of Word Embeddings and Language Models in LMI

Word embeddings and large language models have become essential tools in labour market intelligence, enabling structured analysis of unstructured textual sources such as job advertisements and CVs, and facilitating semantic integration with structured taxonomies like ESCO. The adoption of distributional and contextual representations allows systems to capture subtle relationships between occupations, skills, and qualifications, overcoming the limitations of keyword-based approaches. Static embeddings such as FastText remain valuable for their efficiency and robustness in multilingual and low-resource contexts, while transformer-based architectures like JobBERT and ESCOXLM-R introduce contextualized understanding and fine-grained alignment between languages and taxonomic concepts. These representations support a wide range of downstream applications, including skill extraction, job–skill matching, and cross-country labour market comparison. Overall, embedding-based approaches provide a semantic backbone for modern LMI systems, promoting interoperability, scalability, and deeper insights into evolving skill dynamics across global labour markets.

Part III

Related Work

Chapter 5

Related work in VSM & eXplainable AI

The remarkable progress in Vector Space Models, from static word embeddings to sophisticated Large Language Models, has revolutionized how machines understand and process human language. However, this increasing complexity has created a significant challenge: as these models become more powerful, they also become less interpretable. The transformation of discrete linguistic symbols into high-dimensional vector representations, while effective for capturing semantic relationships, often operates as a "black box," making it difficult to understand the reasoning behind their outputs.

This opacity poses particular problems in high-stakes domains such as finance and labour market intelligence, where understanding *why* a model associates certain terms or makes specific predictions is as crucial as the predictions themselves. For instance, when a financial model identifies emerging market trends or when a labour market system detects skill correlations, stakeholders need to trust and comprehend these insights before acting upon them.

Explainable AI (XAI) addresses this challenge by developing methods and techniques that make the decision-making processes of AI systems transparent, interpretable, and understandable to human users. This chapter explores the intersection of XAI and VSMs, examining how explainability techniques can illuminate the semantic relationships learned by embedding models and provide insights into their internal reasoning mechanisms.

Rather than conducting a traditional paper-by-paper literature review, this chapter adopts a meta-survey approach, analyzing and synthesizing findings from major comprehensive surveys in the XAI field. This methodology is particularly appropriate given the maturity of XAI research and the existence of numerous high-quality surveys that collectively provide extensive coverage of the field's principles, taxonomies, and applications.

5.1 Foundational XAI Methodologies and Frameworks

In their comprehensive survey, Adadi et al. [65] offer a foundational entry point into the field of XAI, aiming to provide a thorough understanding of the key aspects, principles, and methodologies that underpin XAI. The survey is meticulously structured to cover a wide array of dimensions, including the motivations behind the adoption of explainability methods, the diverse applications of XAI across various domains, the methodologies for evaluating approaches, and the strategies employed to implement explainability in machine learning models. While the survey provides a thorough and accessible introduction to the field, its comprehensive scope is slightly marred by the absence of a discussion on the explainability needs of word embeddings and other NLP sub-fields.

Arrieta et al.[66] undertakes a thorough examination of numerous papers, through which they derive a taxonomy grounded in the methodologies employed within these studies. This classification system is developed by considering a variety of dimensions, specifically: the objectives pursued by the research (goal), the intended beneficiaries or audience of the research outcomes (audience they target),

the underlying reasons or necessity for undertaking the research (motivation), the specific methods or approaches utilised in conducting the research (methodology), and how the research findings are conveyed or presented (presentation form). While this survey is comprehensive and the taxonomy provides a structured and nuanced understanding of the existing literature, there is a lack of attention to the vital need for explainable or interpretable models, used to generate word embeddings.

In their seminal work, Burkart et al. [67] present one of the most comprehensive surveys to date on XAI within the context of supervised machine learning. This survey stands out for its depth and breadth, meticulously cataloguing the methodologies, applications, and theoretical concepts of XAI. Through a careful examination of existing literature, similar to Schwalbe et al. [68] the authors provide a curated list of essential definitions and concepts that are central to the discourse on explainable and interpretable AI. However, one area where this survey could be expanded is in addressing the importance of explainability within word embeddings and other sub-fields of NLP.

In Ding et al. [69], the authors develop a multi-dimensional taxonomy designed to facilitate an insightful categorisation of studies within the XAI field. By dissecting the field into distinct categories, the authors aim to provide a granular understanding of the diverse methodologies, application domains, and theoretical concepts, demonstrating the complexity of the field and highlighting the multifaceted nature of explainability in machine learning. Additionally, the paper explores the evaluation metrics for XAI-generated explanations, discussing and categorising these metrics to offer a coherent framework for assessing the quality and effectiveness of explanations. Despite these substantial contributions, the paper does not address the specific need for explainability in the context of word embeddings or other -NLP sub-fields.

Dwivedi et al.[70] primarily concentrates on the technical dimensions of XAI, focusing on the construction of machine learning pipelines that are inherently explainable. It delineates the stakeholders involved in the process and outlines best practices for the creation of such pipelines. Through an in-depth analysis, the authors offer a comprehensive taxonomy of XAI techniques, systematically categorising them based on their functionalities, applications, and intended outcomes. This structured approach facilitates a clearer understanding of the landscape of XAI methodologies, highlighting their diversity and applicability across various domains. Moreover, the paper extends its focus to include XAI software and tools, providing insights into the digital instruments that support the implementation of explainability in machine learning processes. A notable instance of this application-oriented discussion is the mention of TensorBoard, a tool renowned for its capability to project word embeddings, thereby offering a visual interpretation of these complex data structures. Additionally, the authors touch upon dimensionality reduction methods such as t-SNE, which serve as powerful techniques for visualising dense embeddings. Despite these references, it is observed that the discourse on word embeddings and their explicability within the paper is relatively brief. This cursory treatment of word embeddings and the methods for their visualisation and interpretation, such as the singular mentions of TensorBoard and t-SNE, represents a limited exploration of an otherwise crucial aspect of explainability in NLP. In essence, while the paper provides a robust framework for understanding and implementing explainable AI, its engagement with the specifics of word embeddings and their interpretability is somewhat peripheral.

Saeed et al.[71] present a meta-survey on XAI, synthesising findings from a broad array of surveys in the field, offering a panoramic view of the current research landscape. Central to this work is a detailed examination of the challenges and research directions that are pivotal to advancing the domain of XAI. By adopting this meta-analytical approach, the authors aim to distil key insights and identify overarching themes that emerge across multiple studies, thereby providing a comprehensive framework for understanding the complexities and frontiers of XAI research. A notable aspect of this meta-survey is its attention to the machine learning life cycle, echoing the focus seen in similar surveys such as the one conducted by Dwivedi et al. [70]. However, despite the breadth of topics covered, their meta-survey does not address the specific needs for explainability within the domain of NLP, including the critical area of word embeddings.

Similarly, in their meta-study,Schwalbe et al. [68] synthesise the outputs of prior surveys on XAI, to construct a unified taxonomy of XAI methodologies. This endeavour is rooted in the recognition of the fragmented nature of existing classifications and definitions within the field of XAI, which has led to confusion and a lack of cohesion in research efforts. By systematically reviewing and integrating the findings from a wide range of surveys, the authors strive to offer a comprehensive and coherent framework that encapsulates the full spectrum of XAI methodologies. The proposed taxonomy is structured around three primary components: Metrics, Explanators, and Problem Definition, each of which is further delineated into a variety of sub-categories. This hierarchical arrangement ensures that the taxonomy is both detailed and expansive, covering the diverse aspects of XAI research from the measurement of explainability (Metrics) to the agents that generate explanations (Explanators), and the fundamental questions that guide the pursuit of explainability (Problem Definition). Significantly, this meta-study stands out for its acknowledgement of the need for explainability in the domain of word embeddings, an area that has often been overlooked in similar surveys. Within the section dedicated to the "Object of Explanation," the authors highlight three works that have attempted to introduce interpretability into the latent spaces of word embeddings. This mention, albeit brief, signifies an important step towards recognising the complexity and critical importance of making word embeddings and other NLP technologies transparent and understandable.

Very recently, Van et al. [72] present a tertiary review on XAI, synthesising findings from 40 systematic literature reviews between 1992 and 2023. This research systematically categorises the XAI methods and offers insights into which combinations of characteristics and categories have been researched, providing a road-map for future research directions. A notable feature of this review is its systematic documentation of the presence or absence of research in key areas, highlighting gaps in the current landscape. This tertiary review complements earlier works, such as the scoping review by Saeed et al.[71], which focuses more broadly on challenges and research directions in XAI rather than the specific technical characteristics of the methods themselves.

Chapter 6

Related work in Finance domain

Natural language processing has become an increasingly relevant tool in financial research, where text sources such as news, analyst reports, and social media posts carry valuable signals about market behavior. VSM, including both static embeddings such as Word2Vec and FastText and contextual models like BERT, have been widely adopted to represent the semantics of financial language in a numerical form suitable for predictive modeling. These representations capture latent relationships between financial entities, sentiment cues, and economic concepts, enabling the integration of textual information with quantitative market indicators.

In this context, VSMs have been employed for two main purposes. First, they serve as a foundation for financial forecasting, where embeddings extracted from textual data are combined with numerical and temporal features to predict outcomes such as stock price movements, volatility, or trading volume. This multimodal perspective, linking textual and numerical signals, has significantly expanded the modeling capacity of financial prediction systems.

Second, embeddings are increasingly used to study social and behavioral dynamics within online financial communities. The rise of platforms like Reddit and Twitter has produced a rich stream of user-generated content that reflects investors' opinions, collective sentiment, and emerging trends. By representing this text in a shared embedding space, researchers can analyze correlations between sentiment shifts and market fluctuations, providing new insights into retail-driven market phenomena such as the GameStop short squeeze.

The following sections review the main contributions along these two dimensions: (i) VSM-based approaches to financial prediction, emphasizing multimodal and temporal architectures, and (ii) applications of embeddings in social trend analysis and Financial Sentiment Analysis (FSA), particularly focusing on Reddit-based studies.

6.1 Vector Space Models in Financial Prediction

Financial forecasting has long been a core challenge in both academic and industrial research, aiming to predict market trends, asset returns, and risk dynamics to guide trading and investment strategies. Traditionally, forecasting models have relied predominantly on historical market data, such as time series of price and volume, to capture patterns and generate future predictions [73, 74]. However, financial markets are influenced by various exogenous factors, including macroeconomic signals, geopolitical events, investor sentiment, and corporate disclosures, which cannot be fully captured by quantitative indicators alone [75, 76].

The growing availability of heterogeneous financial data sources, encompassing both structured inputs (e.g., financial statements, technical indicators) and unstructured content (e.g., news articles, regulatory filings, social media posts), has fueled interest in multimodal approaches. These models integrate complementary data streams to better reflect the complex informational ecosystem of finan-

cial markets. Recent studies confirm the effectiveness of multimodal fusion in stock prediction across different sectors [77, 78, 79]. Advanced approaches also incorporate causal reasoning and saliency-aware augmentation to improve prediction robustness and exploit macroeconomic signals [80, 81]. By combining multiple modalities, such models can uncover complementary signals, capture latent dependencies, and enhance robustness to market shocks and behavioral dynamics.

Within this multimodal paradigm, Word Embeddings and contextualized language representations play a central role in extracting semantic information from unstructured text. Early works employed static embeddings such as Word2Vec and FastText to encode financial terminology and sentiment features that correlate with market behavior. More recently, transformer-based architectures like FinBERT [82] and Fin-GPT [83] have been applied to financial corpora, providing richer contextual embeddings that capture domain-specific semantics and linguistic nuances. These textual representations are frequently integrated with temporal encoders such as LSTMs [84], or relational modules such as GNNs [85], within end-to-end multimodal frameworks.

Recent literature emphasizes the importance of integrating multiple sources of information to improve predictive performance and model robustness [86]. Hierarchical fusion strategies and large-scale neural architectures have been shown to enhance both accuracy and interpretability [87, 88]. By incorporating embeddings that reflect investor tone, event relevance, or company-specific narratives, these multimodal systems can better anticipate price movements and volatility shifts.

Despite the growing body of work in this area, the literature remains fragmented in terms of architectural choices, data modalities, fusion strategies, and evaluation settings [89, 90, 91].

6.2 Vector Space Models in Financial Social Trend

Stock market forecasting has garnered increasing interest from the research community, leading to the design of various approaches [92, 93], although several methodological and practical challenges remain unresolved [94, 95]. The inherently dynamic and highly non-linear nature of stock prices is driven by different factors, including corporate earnings reports, government policies, actions of influential stakeholders, and expert interpretations of current events [13, 14], as further underlined in recent studies [96, 97].

Despite the rapid advancement of Artificial Intelligence methodologies in financial analysis [98, 99, 100], their predictive accuracy remains constrained in the presence of unexpected or anomalous events [101, 102]. To address these limitations, both academic and industry communities have increasingly explored unstructured textual content, such as news articles and social media posts [103]. Indeed, the increasing availability of unstructured textual data, coupled with the evolution of NLP approaches, led to the development of novel methodologies for capturing investor sentiment in social media. It, however, adds further complexity to stock prediction tasks, which now require the integration and interpretation of high-volume, heterogeneous content to infer user opinions and market-relevant emotions [97]. While traditional news media tend to provide more technical and structured insights, content from social media has demonstrated a stronger influence on public opinion [104, 16]. Dong et al. [105] have demonstrated that news content yields higher predictive reliability on short-term (one-day) horizons, whereas social media exhibits greater utility in capturing market signals over extended periods ranging from two to five days.

Most of the existing approaches to financial market analysis have mainly focused on the processing of news articles [106]. Although Xie et al. [107] provide an extensive open-source evaluation benchmark — including 42 datasets spanning 24 financial tasks and covering eight critical dimensions, namely information extraction (IE), textual analysis, question answering (QA), text generation, risk management, forecasting, decision-making, and bilingual tasks — their focus remains largely constrained to news-based data. This limitation overlooks the variability and complementary perspectives introduced by social media platforms, which have increasingly attracted attention in financial

contexts, as exemplified by the GameStop case [16].

For this reason, the analysis of user sentiment inferred from social textual content has garnered increasing attention in recent years, falling under the umbrella of **Financial Sentiment Analysis (FSA)** [108]. FSA primarily pursues two research objectives: the first focuses on enhancing methodological effectiveness through the design of advanced techniques and the utilization or curation of high-quality human-annotated datasets; the second, which has attracted greater research attention, emphasizes the integration of financial sentiment—either explicitly or implicitly—into downstream applications within financial market prediction and analysis.

The need to explore the relationship between investor sentiment and informed investment decisions, particularly through the integration of sentiment signals, has been emphasized by Qin et al. [109], who incorporate this information into a machine learning framework to rank stocks and generate investment recommendations. However, their approach relies on pre-labeled data sourced from finance-oriented social platforms (e.g., StockTwits), without addressing the inherent challenges of identifying and classifying relevant posts within noisy and unstructured textual environments. In a different vein, Zhuang et al. [110] propose a two-stage method for enabling sentiment-informed decision-making in real-world financial contexts. Initially, a LLaMA-2-13b model is fine-tuned on a manually annotated binary sentiment dataset. The resulting labeled outputs are then used to construct sentiment indices, which serve as input features in a subsequent quantitative investment analysis. While the methodology illustrates the feasibility of leveraging large language models for sentiment extraction, it is constrained by the limited size of the annotated dataset (approximately 200 samples) the reliance on a single large language model for analysis, and the narrow scope of the data source, which is derived from a single investment-focused discussion forum.

Despite recent efforts to build financial-domain LLMs—e.g., FinMA [111] and Fin-GPT [83]—the available models remain relatively small in terms of number of parameters and their performance degrades when applied outside of the specific tasks for which they were optimized. Large foundation models such as BloombergGPT [112], trained with 50 billion parameters on internal Bloomberg datasets, show promising performance, although they are withheld from public release, limiting researchers in assessing domain-specific LLMs at scale in financial applications [113].

As highlighted in the literature, the analysis of opinions expressed in posts is predominantly addressed using FSA techniques, also focusing on a specific use case [16]. This approach is reasonable in the context of news analysis, where positive sentiment is typically associated with good news about companies—encouraging investors to buy shares—while negative sentiment reflects bad news that may prompt investors to sell. However, FSA is not applicable in the context of individual investors' opinions because, as discussed in the introduction, the sentiment of a post is orthogonal to the author's trading intentions. The only work that recognizes the need to measure trading intentions rather than sentiment is Zuang et al. [110], but it essentially employs standard FSA techniques.

Furthermore, the majority of existing approaches focuses on the analysis of news articles, whose content is typically written in a formal language, while other studies investigate sentiment analysis on social media. Although social media content are often characterized by informal and non-standard language, state-of-the-art approaches generally rely on pre-cleaned datasets, as for instance BigData22 [1], ACL18[2], and CIKM18[3].

Recent studies examining users' activity on social media platforms like Twitter, StockTwits, and Reddit have explored the possible impact of these communities on financial market dynamics. These platforms have become powerful mediums for opinion exchange, influencing investment decisions and market trends [114, 115, 116]. Research has shown that financial discussions taking place during periods of increased user activity strongly correlate with stock price movements [117], indicating that social media can influence market volatility through collective sentiment [118, 119] and herding

[1] https://github.com/deeptrade-public/slot

[2] https://github.com/yumoxu/stocknet-dataset

[3] https://github.com/wuhuizhe/CHRNN

behavior [120, 121].

Analyzing the impact of community engagement through the prism of social network metrics gives a better understanding of users' influence on financial market behaviors [122, 123, 116]. Traditional centrality measures, such as degree centrality, betweenness centrality, and PageRank, have been widely used to identify prominent users in social networks. However, not all of these metrics are well-suited for capturing the influential roles on social media platforms due to the inherent structure of the network, which often exhibits non-linear information flows and hierarchical discussion structure [124].

Buntain et al. [123] analyzed network structures of several subreddits to identify distinct social roles within Reddit communities by analyzing users' degree distributions. The study demonstrated that an *"answer-person"* may exert different levels of influence depending on their activity patterns. However, in the framework of financial discussions on Reddit, conventional degree centrality metrics may not capture the full complexity of user influence, as they do not consider the variability of user behavior across different communities. To better capture the dynamic nature of users activity in their study Phang et al. [122] combined degree centralization with other network metrics such as in-degree and out-degree centrality, betweenness, and reciprocity to measure the impact of participation and interaction patterns on user engagement. Their results indicate that a higher level of engagement, manifested by higher scores of inclusiveness and betweenness centrality, allows a user to have a more significant impact on the intentions of others regarding consumption, while a high level of out-degree centrality and a core-periphery structure limit user interaction and lead to the disappearance of this effect.

Recent studies have increasingly focused on integrating content-based analysis, particularly sentiment analysis of textual posts, to define user influence in social network communities. Li et al. [125] focused on user moods and social influence, captured through sentiment analysis on StockTwits to explore the relationship with stock price fluctuations. They demonstrated that sentiment and engagement patterns effectively predict stock trends. Machavarapu et al. [126] have further emphasized the predictive power of sentiment-based measures in financial subreddits, revealing that increased sentiment-driven engagement often correlates with stock price movements, underscoring the need to incorporate engagement metrics into financial market analyses. Long et al. [119] investigated the role of sentiment extracted from Reddit discussions in influencing GameStop's price dynamics during the early 2021 rally. The study focused solely on the predictive power of content extracted from over 10.8 million comments. It evaluated the impact of the sentiments on intraday stock price movements and trading volumes to demonstrate that user discussions play a significant role in market dynamics. The analysis of the reviewed literature reveals that most studies either focus on content-based features alone or utilize network topology measures, overlooking the combined effects of user position within the network and the nature of their contributions. Addressing this gap, Hu et al. [116] integrated the use of PageRank centrality as a proxy for influence with sentiment-driven content features to investigate the impact of Reddit communities like r/wallstreetbets on retail investor behavior, showing that collective actions, such as the GameStop (GME) short squeeze, can create substantial market volatility.

Chapter 7

Related work in Labour Market Intelligence

Machine learning techniques and word embeddings have been extensively applied in Labor Market Intelligence (LMI) for tasks such as job classification, skill extraction, and the generation of synthetic datasets. Online job advertisements represent a rich and dynamic data source, offering valuable insights for real-time labor market analysis and enabling data-driven decision-making [11]. Research has shown that models trained on job vacancy corpora can effectively address a wide range of analytical challenges in this field.

Boselli et al. [11] classify OJAs through Machine Learning (ML) and leverage AI to identify and analyse the unique set of skills required for each profession in the labour market. Colombo et al. [10] develop a set of innovative tools for LMI by applying ML techniques to web vacancies in the IT labour market, allowing them to uncover the nuanced variations in skill demands across various occupations, which may indicate a fragmented job market where specific skills are prioritized for particular roles. They leverage an AI-powered platform to comprehensively analyze the unique blend of skills and qualifications required across various occupations within the dynamic labour market, taking into consideration regional variations, industry trends, educational attainment, and levels of professional experience. Fettach et al. [127] developed a system that reveals the dynamic changes in skill sets and job roles across industries amidst ongoing labor market disruptions from technological advancements and changing societal demands. Malandri et al. [128] develop a Web-based tool for automatically enriching the standard occupation and skill taxonomy (ESCO) with new occupation terms extracted from OJAs. The ability of a profession to be interpreted differently across various markets highlights the importance of understanding the unique characteristics of each market. For instance, Malandri et al. [128] were able to understand that while the ICT labour market in Northern Europe has reached a higher level of maturity compared to the South, it influences how the term *ICT specialist* is perceived and used within those regions. This discrepancy underscores the need for a nuanced approach when interpreting professional titles across markets.

A key application is job classification, where embeddings are leveraged to map job postings onto standardized occupational taxonomies such as ESCO. For instance, **JobBERT** [129] adapts the BERT architecture to job-related text, fine-tuned on occupational corpora to capture semantic nuances in employment contexts and enhance classification performance. Similarly, multilingual models such as the **ESCOXLM-R** model [130] extend XLM-RoBERTa for multilingual job and skill classification aligned with ESCO concepts, enabling better cross-lingual generalization and occupation–skill linking.

Beyond classification, embeddings also play a crucial role in skill extraction and data augmentation. [131] propose a deep learning framework to retrieve relevant skills from job descriptions, addressing the extreme multi-label nature of the task. Similarly, [132] introduce a training strategy

for skill extraction that demonstrates how large language models (LLMs) can improve the coverage and quality of structured labor market data. In a complementary direction, [133] explore the potential of LLMs for zero-shot skill matching, using generative models to align job descriptions and skills without the need for manual annotation.

7.1 Annotated Job Postings Datasets

The development of robust LMI systems is critically dependent on the availability of high-quality, annotated datasets for training and evaluation. The methodologies discussed in the previous sections—such as skill extraction and occupation classification—rely on supervised learning, which in turn requires large-scale, accurately labelled corpora. The landscape of these datasets is diverse, varying significantly in their annotation methodology, granularity, and the types of skills they capture.

Recently, LLMs have demonstrated exceptional proficiency in data generation. Despite this, their generation process is primarily driven by the modelling of statistical correlations between subword tokens. This fact restricts their ability to produce factually accurate text, often leading to hallucinations [134]. To alleviate this issue, several recent approaches propose techniques for knowledge-enhanced synthetic data generation with LLMs, where the contextual information about entities relevant to a text-generation task is converted from a knowledge-based resource into textual form to be included in an LLM prompt[135, 136, 137, 138]. Those approaches, when grounded with trustworthy knowledge related to the domain under consideration, such as the one coming from domain-specific taxonomies or knowledge bases, has proven to enhance the faithfulness to real-world data and the intra-dataset diversity of the generated data, improving the performances of classifiers trained on them [139, 140, 141].

Previous datasets of job postings can be manually annotated, automatically annotated, or synthetic. In some cases, the annotation is inferred directly from the source from which it is extracted. Another difference is that annotation can be on the sentence level or the span level, meaning that each sentence is annotated with the skills it contains instead of the whole job post. Finally, most datasets are annotated only with either soft or hard skills.

Manually annotated Datasets. On the span-level, in [142], 4,863 sentences are annotated with soft skill through crowdsourcing, while [143] and [144] propose respectively 100 and around 200 OJAs manually labelled with hard skills. The dataset presented in [145] comprises around 3,000 OJAs annotated on the sentence level by domain experts with both hard and soft skills. In [146], the authors introduce SKILLSPAN, a dataset containing 14,500 sentences and more than 12,500 annotated spans. They provide construction guidelines developed from three sources and annotated for hard and soft skills by domain experts.

Automatically Annotated Datasets. In [131], Bolha et al. make available a dataset of 20,298 job posts collected from the website mycareersfuture.sg, where the labels come directly from those specified in the job posts as required.

Synthetic Datasets. Both [132] and [133] utilise GPT for the creation of synthetic training data labelled for skill matching. Specifically, in DECORTE [132], the authors prompt GPT-3.5-turbo to generate ten examples for each ESCO skill, whereas in [133], GPT-3.5 is used to produce 40 examples for each ESCO skill. They both generate sentences containing a single skill. An ID field can also be used to reconstruct the job descriptions from the sentences. In SKILLSKAPE [147], the authors propose a generation pipeline ensuring that the skill number varies across job postings, that the skills in the same job posting are correlated, and there is a minimum representation of each skill within the

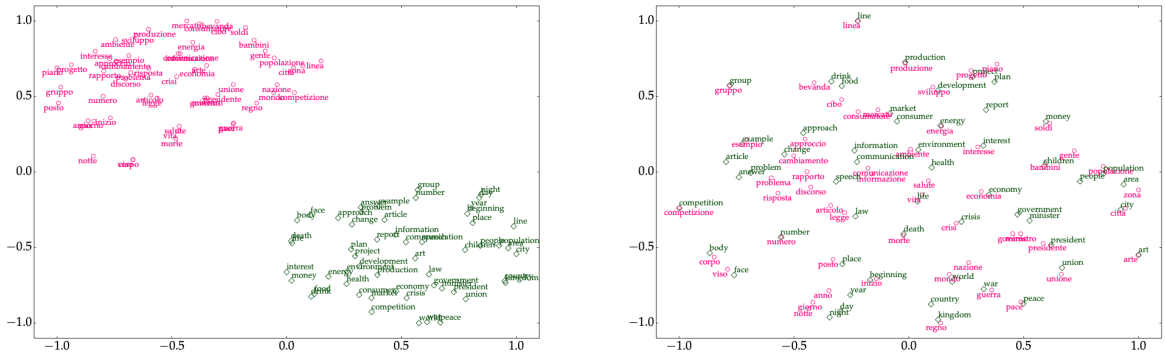


Figure 7.1: An image from [148] depicting unaligned monolingual word embeddings (left) and word embeddings projected into a joint cross-lingual embedding space (right). Embeddings are visualized with t-SNE.

dataset. They employ GPT-2 for generating sentences with more than one skill.

However, a significant limitation persists across these synthetic data generation efforts: they typically produce brief job descriptions containing only one or a few isolated skills. This lack of complexity and the absence of the rich, multi-skill context characteristic of real-world job advertisements represent a substantial discrepancy. This realism gap can consequently hinder the performance and generalisability of models trained solely on such synthetic data for complex downstream tasks like comprehensive skill extraction and job-skill matching.

7.2 Word Embedding Alignment

When word embeddings are trained independently on different corpora, their vector spaces are not directly comparable, even when derived from the same language. This misalignment is exacerbated in multilingual contexts, where embeddings are trained on corpora from different languages, making direct comparisons challenging, as exemplified in Fig. 7.1 from [148]. Without alignment, embeddings from different models cannot be meaningfully compared, limiting their applicability in cross-lingual and cross-domain tasks. Aligning embedding spaces enables meaningful comparisons by mapping different vector spaces into a shared coordinate system. However, this process presents several challenges. First, structural differences arise due to variations in training data, corpus size, and linguistic properties, leading to inconsistencies between vector spaces. Second, selecting appropriate anchor points—words or concepts that serve as reference points for alignment—is critical, as an unsuitable choice can introduce distortions. Finally, the alignment method must balance computational efficiency and accuracy, ensuring that the transformed embeddings preserve semantic relationships while adapting effectively to the target space. Understanding the key challenges in cross-lingual language models is essential for evaluating how state-of-the-art methods address them. Word vectors trained on monolingual data exhibit comparable topological structures across different languages [149]. This observation laid the foundation for early alignment methods, which assumed that embedding spaces could be mapped through a simple linear transformation [149]. However, this assumption has significant limitations. Linguistic variations in morphology, syntax, and semantics introduce complexities that challenge the validity of this hypothesis, making alignment more difficult in practice.

A key challenge in our study arises from our embeddings being trained separately on OJAs' corpora from different countries. Each dataset reflects unique linguistic and contextual nuances, leading to inherently non-comparable vector spaces. Even when the same concept appears across multiple countries, differences in terminology, language use, and data distributions cause variations in learned

embeddings. Consequently, direct comparison of embeddings across countries is infeasible without an alignment mechanism. Various methods have been proposed to achieve this alignment, ranging from offline linear transformations, like [149], to online multilingual embedding adjustments as [150]. These approaches seek to enhance comparability while preserving the integrity of the original vector spaces. A common strategy for cross-lingual alignment involves constructing a seed lexicon—a collection of words with equivalent meanings in both corpora—serving as a reference for mapping embeddings. Within this research domain, several intuitive approaches have emerged. [151] introduced an unsupervised method that exploits the structural similarity of embedding spaces. Instead of relying on bilingual data, they use numerals as anchor points, assuming their meanings remain stable across languages. However, this approach can be sensitive to contextual variations in numeral usage across different corpora. Another notable contribution is the Hierarchical Cross-lingual Embedding Generation (HCEG) method by [152], which eliminates pivot language bias by leveraging linguistic hierarchies. Their approach enhances vocabulary induction, particularly for low-resource languages, though its computational complexity scales with the number of languages involved. A particularly influential method was proposed by [153], who developed an unsupervised alignment technique that does not require parallel data. They learn a linear mapping between embedding spaces by combining adversarial training with Procrustean optimisation. Their method also introduces a novel validation metric and the CSLS similarity measure, achieving results comparable to supervised approaches, even for distant and low-resource language pairs. Among these approaches, SeNSe by [154] offers a distinct perspective. It defines a source model, representing the space to be aligned, and a target model, onto which the source model is mapped. Instead of relying on predefined seed lexicons, SeNSe selects optimal anchors dynamically by identifying words with stable meanings across corpora. It first builds a vocabulary of common words by translating terms and matching them across languages. A semantic similarity score (SNDCG) is then computed to assess alignment quality, retaining only high-scoring anchor pairs while removing duplicates. To ensure a balanced distribution, overly close anchors are filtered out. Finally, these selected anchors are used to learn an orthogonal transformation via Orthogonal Procrustes, preserving semantic relationships while mapping one vector space onto another. Alignment quality is then evaluated through tasks such as Bilingual Lexicon Induction (BLI).

Summary and Limitations of the Background

This background chapter has reviewed the theoretical foundations, methodological paradigms, and application-driven challenges that underpin the research developed in this thesis. Specifically, it has introduced core concepts related to representation learning, word embeddings, graph-based models, sentiment analysis, and multimodal learning, with a focus on their adoption in Labour Market Intelligence and financial applications.

The material presented in this chapter provides the conceptual basis for multiple parts of the thesis. The discussion on distributional semantics, word embeddings, and similarity measures directly supports the methodologies developed in Chapters 9, 10, and 11, where embedding-based representations are employed for keyphrase extraction, synthetic job advertisement generation, and cross-country labour market analysis. The background on social media analysis, sentiment modeling, and network centrality informs the research presented in Chapters 12 and 13, which focus on social trading signals and influence modeling in financial communities. Finally, the overview of multimodal learning and heterogeneous data fusion motivates the systematic survey and taxonomy presented in Chapter 14.

Despite providing a comprehensive overview of existing approaches, the background also highlights several limitations in the current literature. First, many methods rely on static or context-independent representations, which struggle to capture evolving semantics across domains, languages, and time. Second, existing datasets are often limited in size, coverage, or accessibility, constraining reproducibility and large-scale experimentation. Third, most approaches treat textual, numerical, and

relational data in isolation, while real-world economic and financial systems are inherently multimodal and interconnected. Finally, influence modeling in social and financial networks frequently depends on purely structural metrics, overlooking the semantic content and intent expressed by users.

The contributions of this thesis are explicitly designed to address these limitations. By proposing embedding-based, unsupervised methods that adapt to different domains and languages, releasing curated and synthetic datasets, introducing novel tasks and evaluation benchmarks, and advancing multimodal and content-aware modeling strategies, this work moves beyond the constraints identified in the background. As such, the background chapter not only contextualizes the research landscape but also motivates the methodological and empirical innovations developed throughout the thesis.

Part IV

Vector Space Model & eXplainable AI

Chapter 8

eXplainable AI for Word Embeddings: A Survey

In recent years, word embeddings have become integral to Natural Language Processing (NLP), offering sophisticated machine understanding and manipulation of human language. Yet, the complexity of these models often obscures their inner workings, posing significant challenges in scenarios requiring transparency and explainability. This survey conducts a comprehensive review of eXplainable Artificial Intelligence (XAI) strategies focused on enhancing the interpretability of word embeddings. By classifying the existing body of work into six broad categories based on their methodological approaches—a classification that, to our knowledge, does not exist in the literature—we provide a structured overview of current techniques and their characteristics. Additionally, we uncover a noteworthy oversight: a predominant emphasis on interpreting model outputs at the expense of exploring the models' internal mechanics. This finding underscores the necessity of shifting research efforts toward not only clarifying the results these models produce but also demystifying the models themselves. Such a shift is crucial for uncovering and addressing biases inherent in word embeddings, thus ensuring the development of fair and trustworthy AI systems. Through this analysis, we identify key research questions for future studies and advocate for a holistic approach to transparency in word embeddings, encouraging the research community to explore both the outcomes and the underlying algorithms of these models.

This chapter provides a survey presented in [20] that examines the intersection between Explainable Artificial Intelligence (XAI) and word embeddings, aiming to clarify how explainability techniques can be applied to vector space models. We begin by exploring foundational XAI surveys and their relevance to vector space representations, establishing the core principles and motivations behind explainable AI. Subsequently, we analyze comprehensive taxonomies and classification frameworks that have been proposed to organize the diverse range of XAI methodologies. The discussion then shifts to technical implementations and available software tools that facilitate explainability in practice. Furthermore, we consider insights from meta-surveys and tertiary reviews that synthesize the broader XAI landscape, offering a high-level perspective on the field's evolution and current state. Finally, the chapter identifies critical gaps in contemporary XAI research, with a particular focus on the underexplored challenge of word embedding interpretability, and suggests promising directions for future inquiry.

Through this analysis, we aim to provide a comprehensive understanding of how explainability techniques can be applied to vector space models, while highlighting the specific challenges and opportunities in making semantic representations transparent and interpretable.

8.1 Introduction

In the fast-paced world of Artificial Intelligence (AI), the need for models that are not only high-performing but also interpretable and explainable has taken centre stage. As AI technologies are increasingly being integrated into essential areas such as healthcare, finance, and legal systems, the necessity for word embeddings, which transform textual information into numerical formats to reveal semantic relationships, becomes crucial, positioning them at the forefront of NLP techniques. However, the complexity and widespread adoption of these models have rendered their workings increasingly opaque, underscoring the critical need for XAI approaches tailored to this area.

XAI aims to enhance the transparency of AI systems, fostering trust and ensuring accountability in their applications. Within the domain of word embeddings, the quest for explainability is particularly challenging and multifaceted. It involves not only technical efforts to unravel the complexities of the embedding space but also addresses socio-technical issues like bias and fairness in AI outcomes. The role of XAI in word embeddings is critical due to the inherent complexity of human language, making the interpretation and processing of textual data a nuanced task.

This survey aims to provide a thorough analysis of XAI methods in the domain of word embeddings. It outlines the theoretical foundations of word embedding categorises and scrutinises a wide range of XAI techniques for word embeddings, evaluates their practical effectiveness, and discusses ongoing challenges and future research directions. Our goal is to offer a comprehensive overview that will serve as a valuable resource for researchers, practitioners, and policymakers engaged in the development and oversight of transparent and fair AI systems.

8.1.1 Motivation and Contribution

The inception of this survey is propelled by an identified void in the existing literature on XAI within the specific context of word embeddings. Despite the proliferation of XAI research, a focused examination of its application to word embeddings—a cornerstone of NLP technologies—remains conspicuously absent. This gap is not trivial; word embeddings play a pivotal role in interpreting and processing the complexities of human language, a task that is as crucial as it is challenging. The absence of a comprehensive survey in this area hinders the progress toward fully transparent, interpretable, and equitable AI systems.

Our motivation is further fuelled by the increasing reliance on word embeddings in diverse AI applications that affect daily life and critical decision-making processes. As these models become more ingrained in societal functions, the demand for their transparency and explainability escalates. The interpretability of word embeddings presents unique challenges and has a substantial impact on AI outcomes, highlighting the immediate necessity for a dedicated survey. Such a survey is essential to address a significant gap in the literature and to serve as a foundational reference for future research., as we believe this effort will aid in the creation of AI systems that are more transparent, understandable, and fair.

Moreover, this survey is motivated by the belief that advancements in XAI within the domain of word embeddings can drive broader improvements across the AI field. By analysing the complexities of making word embeddings explainable, we aim to uncover new insights into generalisable methods for enhancing model transparency. This effort is not merely academic; it has significant implications for the ethical deployment of AI technologies, ensuring they meet humanity's diverse needs in a fair and responsible manner. Therefore, this survey aims to explore a critically important yet under-explored area, contributing to the development of AI technologies that are both interpretable and innovative.

Fig. 8.1 provides a motivating example that encapsulates the core of our motivation: the necessity of explainable AI for the embedding generation process is as critical as XAI for machine learning tasks. We argue that although NLP tasks have garnered significant attention from the XAI community,

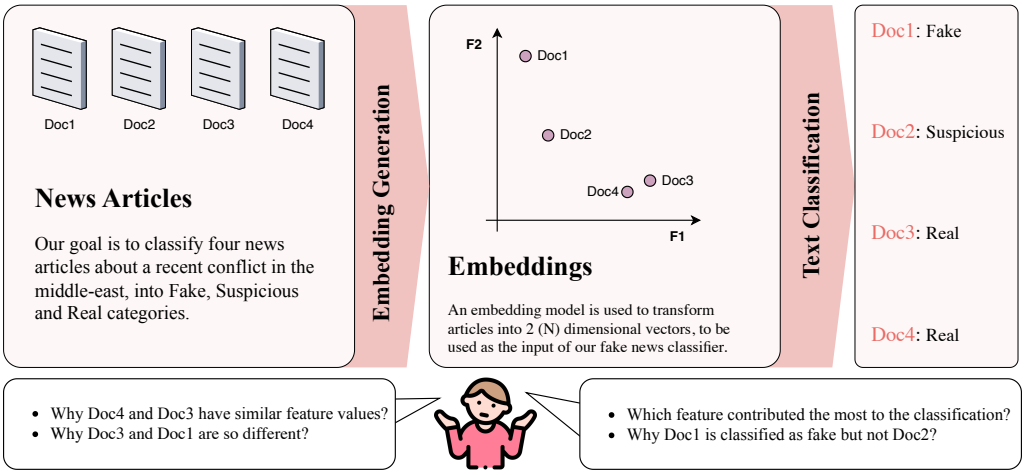


Figure 8.1: Motivating Example: The goal is to classify four news articles into three predefined categories. To achieve this, an embedding model transforms the initial articles into a 2(N)-dimensional space, which is then used as input for the classifier. We argue that the user, developer, or decision maker should be able to question not only the classifier regarding its output but also the embedding model to investigate its inner workings.

explaining embeddings, which serve as the input for nearly all contemporary NLP techniques, have received comparatively less focus. In our survey, we analyse the few studies that offer explainable embeddings or explainers applicable to current embedding methods.

Contributions. Our contribution is two-fold:

- Classification of the current state of the art in word embedding explanation into six distinct groups and discussing their characteristics. To our knowledge, our survey is one of the first works that aims at bridging the gap between XAI and word embeddings.
- Identification of research questions for future studies based on the outcomes of this survey.

8.1.2 Scope

Considering that we encountered a relatively sparse landscape of literature directly addressing this topic, we employed an inclusive methodology to capture a broad spectrum of scholarly contributions in this survey. Consequently, we considered all peer-reviewed works encountered on the topic, without discrimination based on the rank or prestige of the publication venues. This approach resulted in the identification and analysis of 75 papers. By better analysing, we discarded 34 papers which did not have addressed both XAI and word embeddings, resulting in 41 papers. The search terms were identified as combinations of XAI and Word embedding keywords and were further extended, through identification of new keywords from the corresponding papers. The final term sets are XAI (XAI, Explainable Artificial Intelligence, Interpretable AI, Interpretable Artificial Intelligence, Interpretable Machine Learning, IML) and word embedding (Word Vectors, Distributed Representations, Neural Embeddings, Semantic Vectors, Vector Space Models).

We aimed to construct a comprehensive overview that reflects the current state of research, highlighting key methodologies, findings, and insights that collectively advance the discourse on explainability in word embeddings. To maintain a focus on verified and vetted contributions, despite identify-

ing 18 such pre-print articles, we opted not to include them in our analysis. This decision was guided by the desire to ensure that our survey is grounded in peer-reviewed evidence, thus providing a more reliable and rigorous examination of the subject matter.

By delineating our scope in this manner, we seek to offer a solid foundation for understanding the intersection of XAI and word embeddings, while acknowledging the evolving nature of this research domain. Our inclusive yet discerning approach reflects the balance between the breadth of perspectives and the depth of validated knowledge, setting the stage for future explorations to build upon a robust base of peer-reviewed scholarship.

8.2 Word Embeddings & XAI

The NLP field has experienced transformative advancements with the development of word embeddings, pivotal for a wide array of applications. Despite their ubiquity, there remains a significant gap in the literature concerning the interpretability and explainability of these models. Addressing this, our review organises existing research into six categories based on a methodological analysis, focusing on the strategies employed to introduce interpretability. Our classification aims at both highlighting diverse approaches within the domain and also shedding light on potential avenues for future research. To the best of our knowledge, the only survey offering a comparable classification is found in [155]. Unlike our work, however, that study concentrates on Deep NLP models rather than on word embeddings.

8.2.1 Adaptive Embedding Transformation

Description. This group encompasses a range of methodologies aimed at enhancing the interpretability and semantic coherence of word and phrase embeddings. These approaches address the inherent opaqueness of traditional Vector Space Models (VSMs) by introducing novel techniques to uncover and emphasise latent semantic structures within embedding spaces. The methodologies include the development of statistical methods to analyse and quantify the semantic concepts embedded within dense vectors, the application of non-negative and sparsity constraints to maintain interpretability, and the incorporation of compositional semantics to align embeddings more closely with human language intuition. Additionally, visualisation tools are proposed to project embeddings into semantically meaningful sub-spaces, facilitating in-depth analysis and comparison. By transforming pre-trained embeddings through various means—such as denoising auto-encoders, projected gradient descent, and algebraic projection definitions—these methodologies strive to produce embeddings that are interpretable and at the same time, retain or enhance their utility in downstream tasks. Collectively, these approaches contribute to the broader effort of making machine-learned representations more accessible and meaningful to humans, thereby bridging the gap between complex computational models and intuitive human understanding. An example of such methodology is depicted in Fig. 8.2

State-of-the-art. Luo et al. [157] discusses the development of Online Interpretable Word Embeddings (OIWE), focusing on applying non-negative constraints to the Skip-Gram model for learning word embeddings directly from streaming text data. This approach seeks to address the interpretability limitations of traditional word embeddings by ensuring that the learned word vectors maintain non-negativity, making them more interpretable and meaningful. The authors employ projected gradient descent for optimisation, aiming to enhance both the efficiency and interpretability of word embeddings.

Fyshe et al. [158] proposes Compositional Non-negative Sparse Embedding (CNNSE), aiming to overcome the opaqueness of traditional VSMs by introducing interpretability through sparsity and non-negativity constraints. By incorporating the notion of compositionality directly into the learning

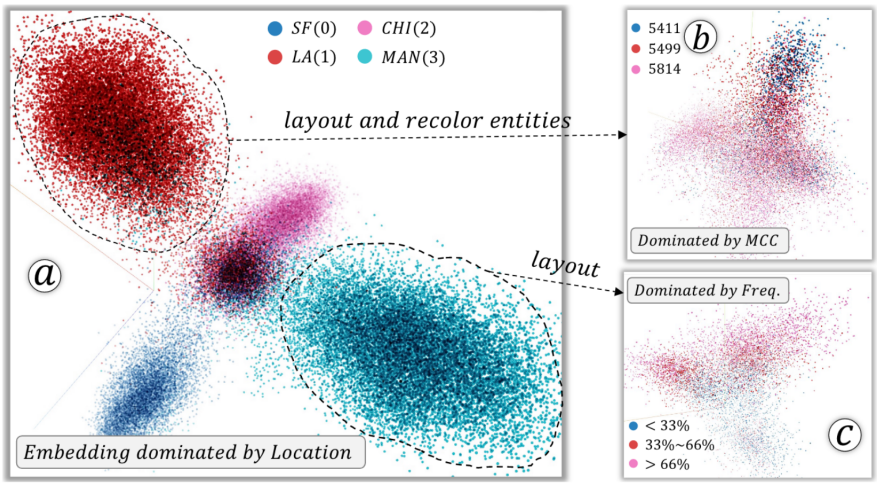


Figure 8.2: An example of Adaptive Embedding Transformation from [156]. The hierarchical organisation within the data embedding reveals a structured partitioning primarily influenced by geographic location, leading to the formation of four distinct clusters corresponding to four cities. This phenomenon is exemplified through merchant embeddings, where, notably, the data from Los Angeles (LA) and Manhattan (MAN) indicate further differentiation within these clusters based on Merchant Category Codes (MCC) and visitation frequency, respectively.

process using the Alternating Direction Method of Multipliers (ADMM) [159], CNNSE yields embeddings with semantically coherent dimensions that align closely with human judgements of phrase similarity. This approach promises more intuitive and semantically rich representations for both individual words and composed phrases.

Yogatama et al [160] transforms dense word vectors into sparse and optionally binary vectors without relying on external information sources. This transformation aims for more interpretable, computationally efficient representations, akin to features used in NLP but discovered automatically from raw corpora. The process involves sparse coding, producing over-complete representations that increase separability and interpretability, with each word having a few active dimensions. The vectors resulting from this process are shown to outperform the original dense vectors on various benchmark tasks.

Rothe et al. [161] introduces DENSIFIER, a method for learning an orthogonal transformation of word embedding spaces, focusing the information relevant to a given task into an ultra-dense subspace. This transformation aims to reduce the dimensionality by a factor of 100 while preserving or even enhancing the task-specific performance and interpretability of embeddings. DENSIFIER operates by identifying an ultra-dense subspace within pre-existing word embeddings (e.g., trained via word2vec or GloVe) without adding or removing information but reorganising it for enhanced task efficiency. The authors validate their approach through experiments on lexicon creation and sentiment analysis tasks, demonstrating that the transformed embeddings retain the original's performance capabilities but with significantly increased efficiency and reduced size.

Rothe et al. [162] introduces a method for enhancing word embedding interpretability by decomposing standard embeddings into orthogonal sub-spaces that represent specific linguistic features like polarity, concreteness, frequency, and part-of-speech, alongside a remainder subspace. This decomposition allows for several operations on word embeddings, such as generating antonyms or neutral sentiment words, effectively extending the traditional analogy operations. To do so, the authors employ an orthogonal transformation, ensuring no information loss from the original embeddings, and utilise lexicon resources for training, focusing on maximising or minimising distances between word pairs based on their linguistic labels. Evaluated across tasks like antonym classification and POS Tag-

ging, the authors claim that their approach demonstrates improved interpretability and performance for various NLP applications.

Andrews et al. [163] explores methods to compress word embeddings, aiming for more compact and interpretable vector representations without significantly compromising performance. Initially, the study employs Lloyd's algorithm [164] to reduce the storage size of GloVe embeddings by a factor of 10 with minimal performance degradation on standard tasks. Utilising the compressed size as a benchmark, they develop a GPU-friendly method to produce sparse, non-negative embeddings.

Jang et al. [165] introduces a novel methodology for identifying the dimensions within word embeddings that significantly denote properties of a word, enhancing the interpretability of word embeddings and enabling property-based meaning comparison. By Analysing components of word embeddings that signal specific properties, they aim to answer questions like "To what degree does a given word possess the property of cuteness?" or "How are two words similar from a specific perspective?". Their approach is validated by correlating the strength of property-signifying components with the degree of prototypicality of target words within categories. The study leverages concepts from category theory and typicality, hypothesising that the strength of significant properties (SIG-PROPS) in word embeddings correlates with a concept's typicality within a category. This is tested using pre-trained Non-Negative Sparse Embedding (NNSE) word embeddings [166] and the Hyper-Lex dataset [167], which provides typicality scores for word-concept pairs. Their results show a strong correlation between the strength of SIG-PROPS and the typicality scores, suggesting that SIG-PROPS can effectively represent essential qualities of a category and thereby confirm the feasibility of extracting property information from word embeddings.

Csenel et al. [168] focuses on uncovering the latent semantic structure in dense word embeddings using a statistical method. It introduces a new conceptual category dataset (SEMCAT) and proposes statistical methods to capture hidden semantic concepts in word embeddings and to measure their interpretability. The approach utilises the Bhattacharya distance metric [169] to analyse the semantic decomposition of word embedding spaces. The results are validated through qualitative and quantitative tests, aiming to quantify interpretability without requiring human effort, as traditionally done in word intrusion tests.

Trifonov et al. [170] creates sparse and interpretable sentence embeddings using auto-encoders. The authors apply sparse coding and model constraints to generate embeddings that are inherently sparse, aiming to enhance interpretability by making the latent space more understandable. This approach includes both post-processing techniques on dense embeddings and direct learning of sparse representations during training. The work also introduces a quantitative metric for assessing embedding interpretability based on topic coherence, demonstrating increased interpretability over dense models without significantly compromising quality.

Similarly, Subramanian et al. [171] proposes SParse Interpretable Neural Embeddings (SPINE) using a denoising k-sparse auto-encoder. This approach aims to transform pre-trained word embeddings like GloVe and word2vec into sparse, interpretable representations. By incorporating a novel learning objective and activation function, SPINE seeks to enhance the interpretability and efficiency of word embeddings. The methodology is validated through large-scale human evaluation and performance comparison on downstream tasks, demonstrating that SPINE embeddings are more interpretable and efficient than the original dense embeddings.

Allen et al. [172] perform an analysis on the underlying structure of word embeddings, focusing on how analogies such as "man is to king as woman is to queen" manifest as linear relationships in embedding space. Their investigation is grounded in a probabilistic definition of paraphrasing, reinterpreted as word transformation, which mathematically describes the analogy "w_x is to w_y." Through this framework, they establish the existence of linear relationships between embeddings, like those produced by word2vec and GloVe, identifying explicit error terms related to these analogical phenomena. Their methodology involves dissecting the paraphrase relationship between word sets, revealing that the addition and subtraction of context words (or embeddings) can approximate other embeddings,

thus illuminating the process by which certain word transformations preserve linear analogical relationships.

Molino et al. [173] proposes Parallax, a visualisation tool for embedding spaces. It offers traditional methods like PCA and t-SNE, along with a novel approach allowing users to define projection axes through algebraic formulas. This method aims at enhancing interpretability by projecting embeddings into semantically meaningful sub-spaces, enabling detailed analysis and comparison across different models or corpora. The proposed tool is demonstrated through case studies including bias detection and polysemy analysis, showcasing its flexibility for various analytical tasks.

Templeton et al. [174] outlines the creation of inherently interpretable sparse word embeddings through a process called sparse coding. This process transforms pre-trained dense word embeddings into sparse embeddings where each dimension represents a natural language word or specific grammatical concept. The approach aims to make word embeddings more interpretable by ensuring that each dimension of these sparse vectors corresponds to a human-understandable concept, thus enhancing the ability to understand and analyse the semantic and grammatical properties embedded within these vectors.

Garcia et al. [175] introduces Intermediate Entity-based Sparse Interpretable Representation Learning (ItsIRL), aiming to improve the predictive performance of Interpretable Entity Representations (IERs) without sacrificing their interpretability. This is achieved by incorporating an intermediate interpretable layer that outputs entity types, which are then decoded into dense layers for downstream predictions. The authors claim that ItsIRL addresses the challenge of maintaining interpretability after fine-tuning, which typically degrades the semantic meaning of entity types learned during pre-training.

Zheng et al. [156] proposes EmbeddingTree for hierarchical exploration of entity features in embedding spaces. It involves a Gaussian Mixture Model (GMM)-based algorithm that uses entity features to split the embedding space, selecting the best feature for splitting through Bayesian Information Criterion (BIC) approximation. This process creates a tree structure that organises embeddings according to hierarchical entity features, aiding in the interpretability and exploration of embeddings. Additionally, an interactive visualisation tool is developed to visualise and explore the hierarchical structure and the embeddings within it.

8.2.2 Semantic Concept Alignment

Description. This group of methodologies aim to enhance the interpretability of word embeddings by aligning them with human-understandable semantic concepts. This approach typically involves transforming the representation space of pre-trained embeddings or modifying the objective function of embedding learning algorithms to incorporate semantic differentials or predefined conceptual dimensions. An example of such methodology is depicted in Fig. 8.3.

State-of-the-art. Mathew et al. [177] proposes a method for enhancing the interpretability of pre-trained word embeddings through the adoption of semantic differentials. This approach transforms existing embeddings into a new space with dimensions defined by polar opposites, such as "cold-hot" or "soft-hard," using sets of polar opposites provided by an external oracle. The transformed embeddings are then evaluated on various downstream tasks to demonstrate that adding interpretability does not compromise performance.

Csenel et al. [176] proposes a method for imparting interpretability to word embeddings while preserving their semantic structure. It involves modifying the objective function of the embedding learning algorithm, encouraging vectors of words semantically related to predefined concepts to align along specified dimensions. This is achieved using concepts derived from Roget's Thesaurus [178], aiming for embeddings where dimensions correspond to human-understandable concepts. The approach seeks to make the dense, typically non-interpretable embeddings into interpretable ones without sacrificing their performance on standard NLP tasks.

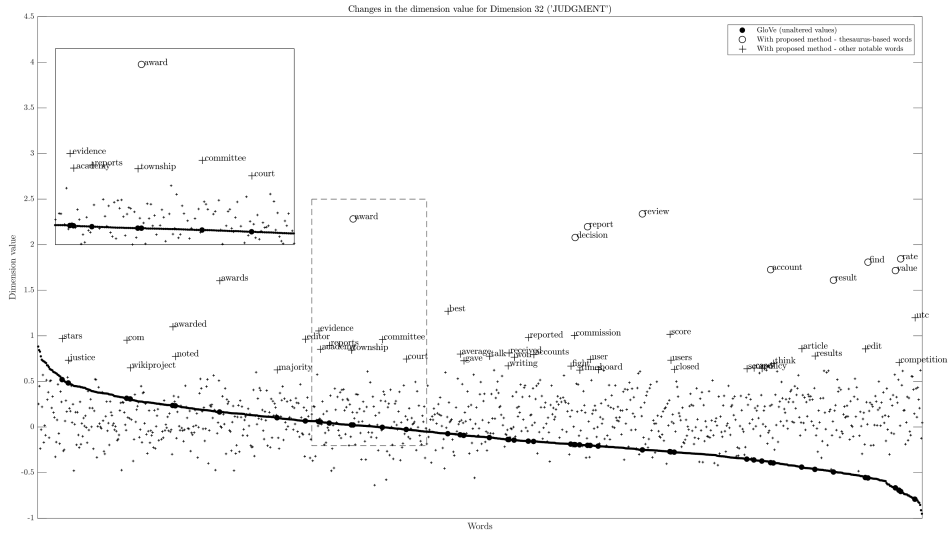


Figure 8.3: An example of Semantic Concept Alignment from [176]. The analysis focuses on the most frequent 1000 words, initially ranked according to their positions in the dimension of the original GloVe embedding. This study further contrasts these positions by indicating the corresponding values of these words when aligned with the dimension of the embedding.

Csenel et. al [179] proposes BiImp, a bidirectional imparting algorithm for improving the interpretability of word embeddings. This method enhances word embeddings by encoding different concepts in both positive and negative directions of each embedding dimension. The authors address the limitations of unidirectional imparting by utilising the full capacity of the embedding space, offering a more efficient use and increased encoding flexibility. The method is evaluated by demonstrating its ability to maintain semantic task performance while enhancing interpretability and reducing gender bias in word embeddings.

Engler et al. [180] introduces SensePOLAR, a technique that enhances the interpretability of pretrained contextual word embeddings by differentiating word senses. It extends the POLAR framework introduced by [177], utilising semantic differentials (scales between two antonyms) to create a sense-aware space. This approach allows for distinguishing between different senses of a word by selecting polar sense dimensions from an oracle (like WordNet), generating sense embeddings, constructing a polar sense space, and transforming original embeddings to this space for interpretation.

8.2.3 Semantic Enrichment through Knowledge Integration

Description. This group of methodologies focuses on enhancing word embeddings by integrating external, structured knowledge sources. The aim is to improve semantic coherence and interpretability, making the embeddings not only richer in semantic content but also more aligned with human understanding. An example of such methodology is depicted in 8.4.

State-of-the-art. Murphy et al. [166] develops Non-Negative Sparse Embedding to create sparse, interpretable word embeddings from large text corpora. This method applies matrix factorisation with non-negativity constraints to generate embeddings where each word is represented by a few active, meaningful dimensions. NNSE emphasises sparsity and interpretability, contrasting with denser, less interpretable embeddings like those from SVD. The interpretability is demonstrated through a word intrusion detection task, showing NNSE's ability to produce semantically coherent dimensions that align with human cognitive models.

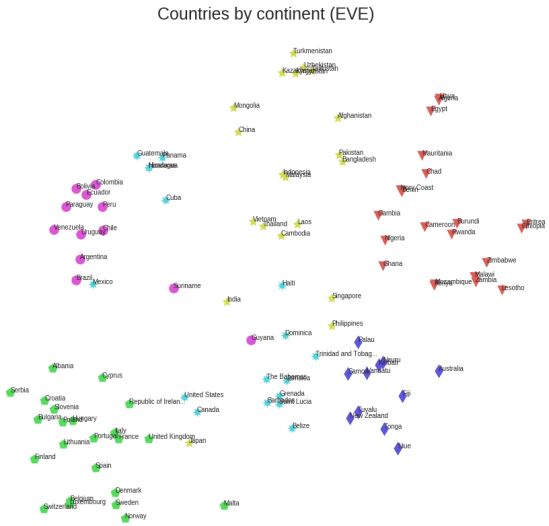


Figure 8.4: The visualisations of model embeddings, presented in [181] as an example of Semantic Enrichment through Knowledge Integration group, is designed to showcase the model's capability in the clustering task, particularly for categorising items based on the 'topical type' from Country to Continent.

Faruqui et al. [182] proposes a technique called retrofitting, which refines pre-trained word embeddings by leveraging the relational information from semantic lexicons like WordNet [183], FrameNet [184], and the Paraphrase Database [185]. The retrofitting process operates as a post-processing step, making it applicable to any set of pre-trained vectors, regardless of their initial training methodology. This method works by creating a graph-based structure where words and their semantic relationships form nodes and edges, respectively. By running belief propagation on this graph, retrofitting adjusts the word vectors to make semantically related words more similar in the vector space, while also keeping the updated vectors close to their original distributionally-derived representations. Based in the provided evaluation, the approach improves the quality of word embeddings across a variety of standard lexical semantic tasks and languages, outperforming existing methods for incorporating semantic lexicon information into word vector training.

Zhao et al. [186] addresses the issue of gender stereotypes in word embeddings by introducing a novel training method for generating gender-neutral embeddings, specifically a Gender-Neutral variant of GloVe (GN-GloVe). This approach aims to isolate gender information into specific dimensions of word vectors, thereby preserving gender neutrality in the remaining dimensions without losing the embeddings' overall functionality. The authors state that unlike previous methods that either completely remove gender information or require external classifiers to identify gender-neutral words, GN-GloVe integrates the identification of gender-neutral words within the training process itself, thereby avoiding error propagation. The methodology involves adjusting the training objective to ensure that gender information is confined to predetermined dimensions, making the resulting word embeddings more interpretable and less prone to perpetuating gender biases.

Panchenko et al. [187] introduces a technique to improve the interpretability of word sense embeddings [188] by linking them to synsets in a lexical resource like BabelNet [189]. This method aims to bridge the gap between the adaptivity of corpus-driven models and the interpretability of lexicon-based approaches, using AdaGram [190] sense embeddings as a case study. By aligning sense vectors with

synsets, the approach seeks to enhance the usability of embeddings in NLP applications by providing a more understandable semantic framework.

Jha et al. [191] leverage category theory and categorical knowledge in the biomedical domain to enhance the interpretability of word embeddings. It involves learning a transformation matrix that projects pre-trained word embeddings into a new space, where embeddings become interpretable by aligning with human-defined categories. This approach aims to uncover the hidden conceptual meanings of individual dimensions within the embeddings while retaining their original expressive features.

Lauretig et al. [192] develops Bayesian word embeddings with automatic relevance determination priors, enhancing interpretability and allowing for the incorporation of prior knowledge into embeddings. To achieve this, the approach treats word embeddings as Bayesian latent variable models and applies ideal point modelling techniques for dimension anchoring, enabling their use in regression analysis.

Qureshi et al. [181] proposes a method called EVE (Explainable Vector Based Embedding Technique) that creates word embeddings that are inherently interpretable by utilising the structure of Wikipedia. This process seeks to ensure that related words share similar article and category associations, making the embeddings more understandable and semantically rich.

Bodell et al. [193] uses informative priors on word types expected to discriminate on specific dimensions (e.g., gender) to enhance interpretability and connect to research interests, i.e. explicitly incorporating domain knowledge into the embedding process.

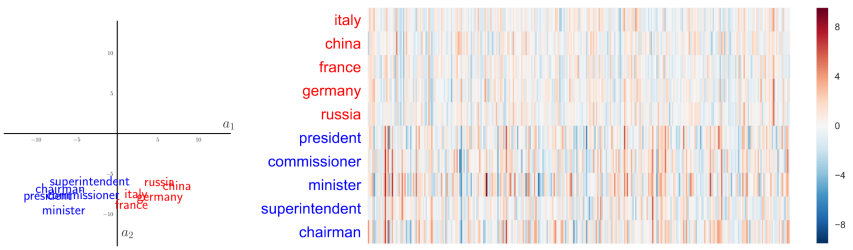
Tang et al. [194] proposes Sparse Variational Auto-encoder-Based Interpretable Bimodal Word Embeddings (BIWE). This approach combines textual and visual information to generate word embeddings that are both interpretable and effective for downstream tasks. It leverages sparse VAE to learn interpretable representations, aiming to improve the semantic understanding of words by integrating visual cues into the embeddings. The authors claim that their proposed method addresses the challenge of enhancing word embedding interpretability without compromising performance in downstream tasks.

Li et al. [195] incorporates human knowledge into data embeddings to improve pattern significance and interpretability. The core idea involves two main steps: externalising tacit human knowledge as explicit sample labels and adding a classification loss in the embedding network to encode samples' classes. This approach aims to bring samples of the same class with similar data features closer in the projection, resulting in more compact and class-consistent visual structures. An embedding network with a customised classification loss implements this idea, integrated into a visualisation system to support flexible class creation and pattern exploration.

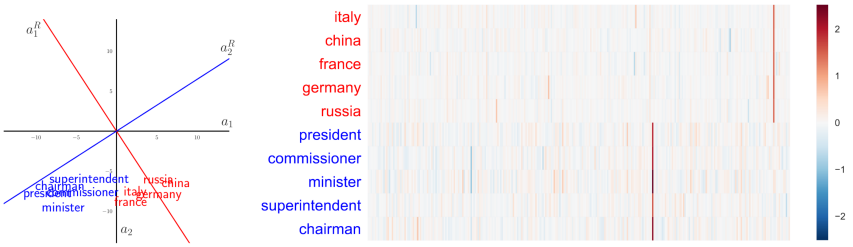
8.2.4 Embedding Rotation

Description. This category includes methodologies aimed at reorienting word embedding spaces to make their dimensions more interpretable. By applying mathematical transformations, such as rotations, these techniques seek to restructure embedding matrices so that their components align more closely with human understandable concepts or linguistic structures. The goal is to enhance the utility of embeddings for linguistic analysis and NLP applications by making the semantic relationships within the embedding space clearer and more accessible. An example of such methodology is depicted in Fig. 8.5.

State-of-the-art. Park et al. [196] proposes a method for improving the interpretability of pre-trained word embeddings, like GloVe and word2vec, through rotation algorithms without sacrificing performance. Their method involves applying four rotation algorithms (Quartimax, Varimax, Parsimax, and Factor Parsimony) to word vector representations to make the dimensions interpretable. This approach maintains the original vectors' expressive power and is validated using various NLP tasks, including a word intrusion task. The rotated vectors are shown to be more interpretable and retain



(a) SG word projected to $\{a_1,a_2\}$ and visualization of the vectors in 300 dimensions



(b) Rotated word vectors in $\{a_1^R,a_2^R\}$ and visualization of the vectors in 300 dimensions

Figure 8.5: Park et al.[196] provides an insightful analysis through the overview of rotating word vector dimensions. This analysis encompasses the comparison between (a) unrotated and (b) rotated skipgram word vectors, depicted in a 2-D projected embedding space facilitated by PCA on the left, alongside a visualisation in the original 300-dimensional space on the right. The colour coding within these visualisations—red for countries and blue for positions—serves as a semantic guide.

or improve upon the original vectors' performance in tasks like word similarity, semantic/syntactic analogy, and classification tasks. This method serves as a post-processing step that can be applied to any word vector model, improving interpretability without additional training or increasing model complexity.

Zobnin et al. [197] provide a method for enhancing the interpretability of word embeddings through orthogonal rotations, specifically applying SVD to word embedding matrices. This approach is aimed at making the components of the embeddings more interpretable and stable across re-training sessions. By studying different models for the Russian language, the research demonstrates that applying canonical orthogonal transformations can reveal meaningful dimensions within the embeddings, potentially making them more useful for linguistic analysis and NLP applications.

Dufter et al. [198] introduces analytical methods for making ultra-dense word embeddings interpretable through rotation. It examines three methods: Densifier [161], linear SVMs, and a new method called DensRay. DensRay, an advancement over Densifier, offers a closed-form, hyper-parameter-free solution for creating interpretable word spaces by optimising the embedding's orientation. This is demonstrated through tasks like lexicon induction and set-based word analogy, providing both a quantitative evaluation and qualitative insights into the utility of interpretable word spaces for applications such as debiasing embeddings.

Ethayarajh et al. [199] investigates the representation of word relationships not just as linear translations but through orthogonal and linear transformations in the embedding space. This perspective challenges the traditional view that word relationships, such as gender differences or tense changes, are merely vector offsets (e.g., "king" to "queen" represented as "king" - "man" + "woman"). The authors argue that word relationships can also be accurately modelled as rotations and reflections using orthogonal matrices or even more generally as linear transformations, achieving comparable or superior performance to traditional vector arithmetic in analogical reasoning tasks. This finding suggests a broader understanding of how word relationships can be encoded in embeddings and hints at the potential mechanisms through which downstream models, including those based on attention

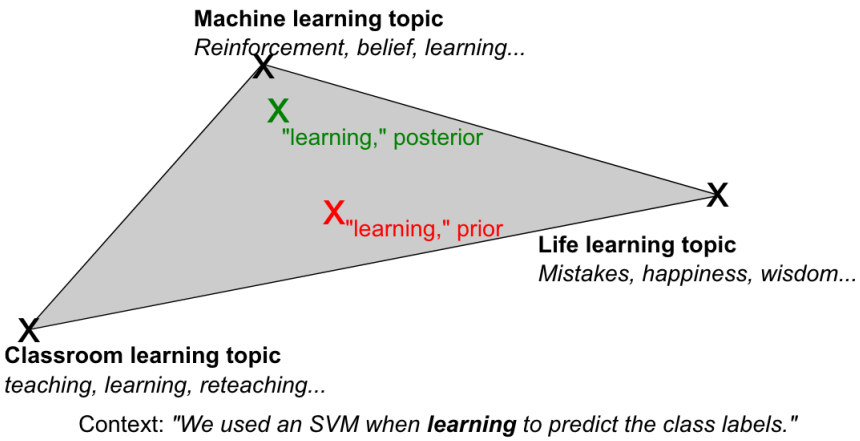


Figure 8.6: As presented in [200] (Topic Modelling Techniques), this Fig. displays the construct of mixed membership word embeddings. The prior word type embeddings, alongside the posterior word token embeddings are portrayed, and formulated as convex combinations of topic embeddings.

mechanisms like Transformers, might learn and represent linguistic relationships.

8.2.5 Topic Modelling Techniques

Description. This group of methodologies employs topic modelling techniques to create word embeddings that are inherently more interpretable. By breaking down words into sparse distributions over topics or senses, these approaches seek to uncover the multiple semantic facets of words, making their meaning clearer and more nuanced. An example belonging to this category is depicted in Fig. 8.6.

State-of-the-art. Reisinger et al. [201] proposes a multi-prototype approach that, unlike traditional models represents a word with a single prototype vector. It employs clustering to create multiple sense-specific vectors per word, thus better handling homonymy and polysemy. The proposed model adapts to the context, selecting the most relevant vector representation for a word based on the surrounding context, thereby offering a more dynamic and accurate representation of word meanings. The paper demonstrates through experiments that this multi-prototype model correlates better with human judgements of semantic similarity, both in isolated words and in context than both single-prototype and exemplar-based approaches. The methodology includes unsupervised word sense discovery by clustering contexts in which a word appears, and then calculating semantic similarity using these clusters.

Pelevina et al. [202] introduces an effective approach for learning word sense embeddings that can generate a sense inventory from pre-existing word embeddings through the clustering of ego-networks of related words. Unlike methods that directly learn sense representations from corpora or depend on lexical resources, their approach offers a novel means of inducing sense inventories and integrates a Word Sense Disambiguation (WSD) mechanism. This allows for the contextual labelling of words with learned sense vectors, enabling various downstream applications. Their experiments demonstrate that the method achieves performance comparable to state-of-the-art unsupervised WSD systems.

Foulds et al. [200] outlines a model-based method for training interpretable, corpus-specific word embeddings for computational social science. It leverages mixed membership representations, Metropolis-Hastings-Walker (MHW) sampling, and noise-contrastive estimation (NCE). The approach aims to address the limitations of big data reliance and lack of interpretability in traditional word embeddings, proposing a solution that performs well even on smaller datasets. The authors claim that their model

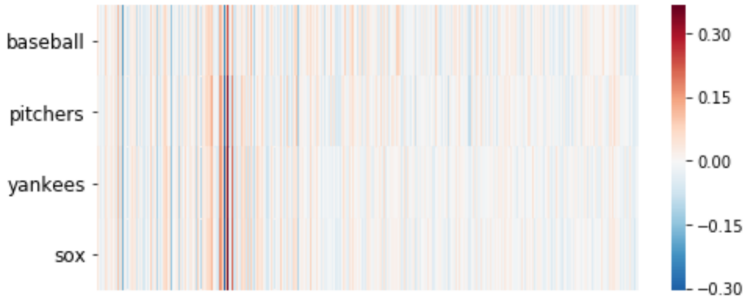
enhances the interpretability and relevance of embeddings for computational social science applications by incorporating mixed membership models and efficient sampling techniques.

Snidaro et al. [203] discusses building explainable word embeddings by leveraging distributional semantics, which posits that words acquire meaning from the contexts in which they appear. The approach emphasises the construction of explicit, interpretable vectors that represent semantic features as dimensions. The proposed methodology involves a detailed examination of word co-occurrences and the application of statistical methods to derive embeddings that can be directly associated with linguistic contexts.

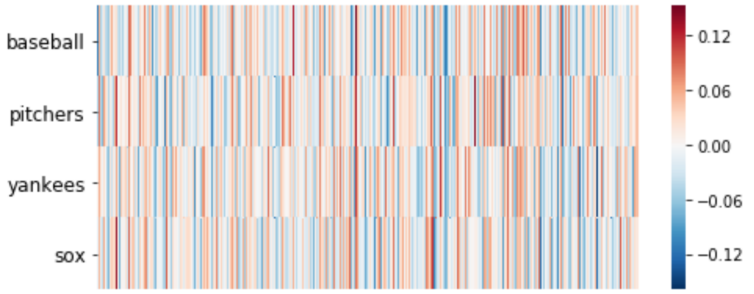
Panigrahi et al. [204] introduces Word2Sense, a generative model for producing sparse, interpretable word embeddings that capture multiple senses of a word. This method represents words as sparse probability distributions over senses, improving upon dense embeddings by making the semantic structure of words clearer and more accessible. The approach leverages LDA (Latent Dirichlet Allocation) to identify fine-grained senses from large corpora, aiming for embeddings that are both interpretable and effective across various NLP tasks.

8.2.6 Others

Upon evaluation, this category of works diverges from the established criteria of the previous categories. Despite this divergence, these works stand out for their introduction of innovative and effective methodologies, making distinctive contributions to the field. An illustrative example from this category is presented in Fig. 8.7.



(a) Word vectors from $\mathbf{W}^{\text{SVD}}$



(b) Word vectors from $\mathbf{W}^{\text{SGNS}}$

Figure 8.7: Comparison of the representations of a set of words related to a popular sport in two different embedding models, as presented in [205]. The models (denoted Model 1 and Model 2), are illustrated where rows correspond to embedding dimensions.

Zhang et al. [206] aims to improve word embeddings' interpretability by integrating definition and usage generation. The novel framework leverages an encoder-decoder architecture to produce context-sensitive definitions of words. It introduces usage modelling, exploring the potential of embeddings

to generate example sentences, aiming for a direct and explicit expression of the semantics embedded in the word vectors. This is further extended into a multi-task learning setting, combining definition and usage modelling to enhance performance and interpretability simultaneously.

Shin et al. [205] proposes a method for interpreting word embeddings through eigenvector analysis. This approach analyses the eigenvectors of word embeddings obtained by truncated SVD of the Positive Pointwise Mutual Information (PPMI) matrix and compares this with the analysis of embeddings trained using Skip-Gram with Negative Sampling (SGNS)[1]. The study employs Random Matrix Theory to investigate the distributions and Inverse Participation Ratios (IPR) of eigenvectors, aiming to identify significant semantic features within the embeddings without imposing additional constraints or post-processing steps on the embeddings.

Opitz et al. [207] aims at improving the interpretability of SBERT (Sentence-BERT) embeddings [208] by decomposing them into semantically structured sub-embeddings (S3BERT). This approach leverages Abstract Meaning Representation (AMR) [209] metrics to guide the decomposition, aiming to make the embeddings more interpretable by emphasising specific semantic sentence features like semantic roles, negation, and quantification. The method includes a consistency objective to prevent catastrophic forgetting and preserve the effectiveness and efficiency of the neural embeddings.

8.2.7 Towards Granular Explanations for Word Embeddings

Granular computing is a computational paradigm that deals with the processing of complex information entities called "granules," which can be any meaningful collections of data, such as words, phrases, or sub-phrases in the context of word embeddings (See [210, 211]) The primary benefits of granular computing include enhanced interpretability, improved computational efficiency, and the ability to handle uncertainty and imprecision in data [212]. By breaking down complex information into finer granules, granular computing enables more detailed and nuanced analysis, leading to more precise and insightful explanations. To determine which categories of the reviewed technologies are inherently suitable for granular explanations (e.g., explanations at the word or sub-word level), we analyse the mentioned categories below:

Adaptive Embedding Transformation. This category involves enhancing interpretability and semantic coherence by introducing techniques that uncover latent semantic structures within embedding spaces. This includes statistical methods, non-negative constraints, and compositional semantics. These methods often aim to make embeddings more interpretable on a fine-grained level by transforming them in a way that aligns more closely with human language intuition.

Semantic Concept Alignment. These methodologies enhance interpretability by aligning embeddings with human-understandable semantic concepts, often by transforming the representation space or modifying the objective functions to incorporate semantic differentials. This category is well-suited for granular explanations as it directly links embedding dimensions to specific semantic concepts.

Semantic Enrichment through Knowledge Integration. This category focuses on integrating external structured knowledge sources to improve semantic coherence and interpretability. By incorporating knowledge bases, these methods enhance the richness and human-alignment of embeddings, which can provide detailed insights at a granular level.

Embedding Rotation. Involves rotating the embedding space to make certain semantic features more explicit. This can help in identifying and explaining specific dimensions and their semantic roles within the embedding space. Such methods can be particularly effective for granular explanations as they often result in a more interpretable geometric structure.

Topic Modelling Techniques. Uses topic modelling to create interpretable embeddings by breaking down words into sparse distributions over topics or senses. This approach inherently provides granular insights into the different senses or topics associated with words, making it suitable for detailed explanations.

Others. This category includes innovative methodologies that do not fit neatly into the above categories but still contribute to the field. They often introduce unique ways of enhancing interpretability and could include methods that provide fine-grained explanations depending on their specific approaches.

We argue that the categories most inherently suitable for granular explanations are **Semantic Concept Alignment**, **Topic Modelling Techniques**, and **Embedding Rotation**. These categories focus on aligning embeddings with understandable concepts, breaking down words into multiple senses or topics, and transforming the embedding space to highlight specific features, respectively, all of which provide detailed insights at the word or sub-word level.

Conclusion. The section categorises existing methodologies for explaining word embeddings into six main families: Adaptive Embedding Transformation, Explainable Neural Embedding Models, Post-hoc Interpretation Techniques, Structured Embedding Learning, Interpretable Embedding Visualisation, and Hybrid Methods. These approaches focus on enhancing the interpretability and semantic coherence of word embeddings by employing techniques such as statistical analysis, non-negative and sparse constraints, visualisation tools, attention mechanisms, probing tasks, hierarchical structures, and hybrid models. These methodologies provide diverse approaches to enhancing the interpretability of word embeddings. By classifying and reviewing these methods, this section highlights the current state of research and points to potential directions for future work, aiming to bridge the gap between complex computational models and intuitive human understanding.

8.3 Results

Table 8.1 provides a compilation of research findings, aligned with the evaluation criteria similar to the survey done by [213]. These criteria include Benchmark, Evaluation, Dataset, Code, and Type of Explanation, offering a structured approach to assessing the works under review. Each criterion is defined as follows:

Benchmark. This criterion assesses whether the authors have established a benchmark by comparing the efficacy of their proposed method against existing methods. The establishment of a benchmark is crucial for understanding the relative performance and advancements introduced by the new method.

Evaluation. This aspect examines the thoroughness of the method's evaluation process.

Dataset. This criterion distinguishes between the use of public and private datasets in the research.

Code. Availability of the code is another critical factor, as it enables peer verification, facilitates further research, and enhances the transparency and reproducibility of the results.

Explanation Type. The type of explanation refers to the methodological approach used to enhance interpretability. "Post-hoc" explanations aim to shed light on the embeddings generated by a black-box model (e.g. GloVe) and make them interpretable. In contrast, "ante-hoc" explanations involve methods that generate embeddings that are inherently interpretable.

Table 8.1: *(Benchmark)* → No: ☐, Provided: ■ ; *(Evaluation)* → Performed: ✖, Not performed: 🔍 ; *(Dataset)* → Not mentioned: 👁‍🗨, Public dataset: 🔓 ; *(Code)* → Not provided: 🔑, Provided : git ; *(Explanation type)* → Ante-hoc: ◀, Post-hoc: ▶ ; *(Approach)* → Count-based: ⁘, Prediction-based: 🪄 ; *(Architecture)* → Contextual: ⌛, Non-Contextual: ⧖

Category	Paper	Year	Benchmark	Evaluation	Dataset	Code	Exp. Type	Approach	Architecture	Inference
Adaptive Embedding Transformation	Fyshe et al.[158]	2015	■	🔍	🔓	🔑	▶	⁘	⧖	▶
	Luo et al. [157]	2015	■	✖	🔓	git	▶	🪄	⧖	Skip-Gram
	Yogatama et al. [160]	2015	■	🔍	🔓	git	▶	🪄	⧖	▶
	Andrews et al. [163]	2016	■	🔍	🔓	🔑	▶	🪄	⧖	Autoencoder
	Rothe et al. [161]	2016	■	🔍	🔓	🔑	◀	🪄	⧖	▶
	Rothe et al. [162]	2016	■	🔍	🔓	🔑	◀	🪄	⧖	▶
	Jang et al. [165]	2017	☐	🔍	🔓	🔑	◀	🪄	⧖	▶
	Subramanian et al. [171]	2017	■	🔍	🔓	git	▶	🪄	⧖	Autoencoder
	Trifonov et al. [170]	2018	☐	🔍	🔓	🔑	▶	🪄	⧖	RNN
	Csenel et al. [168]	2018	☐	🔍	🔓	🔑	▶	🪄	⧖	▶
	Allen et al. [172]	2019	☐	✖	👁‍🗨	🔑	◀	🪄	⧖	Skip-Gram
	Molino et al. [173]	2019	☐	🔍	🔓	git	◀	🪄	⧖	▶
	Templeton et al. [174]	2021	■	🔍	🔓	🔑	▶	🪄	⧖	▶
	Garcia et al. [175]	2022	■	🔍	🔓	git	▶	🪄	⌛	BERT
	Zheng et al. [156]	2023	☐	🔍	🔓	🔑	◀	🪄	⧖	▶
Semantic Concept Alignment	Mathew et al. [177]	2020	■	🔍	🔓	git	▶	🪄	⧖	▶
	Csenel et al. [176]	2021	■	🔍	🔓	git	▶	🪄	⧖	▶
	Csenel et al. [179]	2022	■	🔍	🔓	git	▶	🪄	⧖	Skip-Gram
	Engler et al. [180]	2023	■	🔍	🔓	git	▶	🪄	⌛	BERT
Semantic Enrichment through Knowledge Integration	Murphy et al. [166]	2012	■	🔍	🔓	🔑	▶	⁘	⧖	▶
	Zhao et al. [186]	2015	■	🔍	🔓	git	▶	🪄	⧖	▶
	Faruqui et al. [182]	2015	■	🔍	🔓	🔑	◀	🪄	⧖	▶
	Panchenko et al. [187]	2016	■	🔍	🔓	🔑	◀	🪄	⌛	Skip-Gram
	Qureshi et al. [181]	2017	■	🔍	🔓	git	▶	⁘	⧖	▶
	Jha et al. [191]	2018	■	🔍	🔓	🔑	▶	🪄	⧖	▶
	Lauretig et al. [192]	2019	☐	🔍	🔓	git	▶	🪄	⧖	▶
	Bodell et al. [193]	2019	■	✖	🔓	git	▶	🪄	⧖	▶
	Tang et al. [194]	2021	■	🔍	👁‍🗨	🔑	◀	🪄	⧖	VAE
	Li et al. [195]	2022	■	🔍	🔓	git	◀	🪄	⌛	RNN
Embedding Rotation	Park et al. [196]	2017	■	🔍	🔓	git	◀	🪄	⧖	Skip-Gram
	Zobnin et al. [197]	2018	☐	✖	🔓	🔑	◀	🪄	⧖	feed forward
	Dufter et al. [198]	2019	■	🔍	🔓	git	◀	⁘	⧖	▶
	Ethayarajh et al. [199]	2019	☐	🔍	👁‍🗨	🔑	◀	🪄	⧖	feed forward
Topic Modeling Techniques	Reisinger et al. [201]	2010	☐	🔍	🔓	🔑	◀	⁘	⌛	▶
	Pelevina et al. [202]	2016	■	🔍	🔓	🔑	▶	🪄	⧖	CBOW
	Foulds et al. [200]	2018	■	🔍	🔓	🔑	◀	🪄	⌛	Skip-Gram
	Snidaro et al. [203]	2019	☐	🔍	🔓	🔑	▶	⁘	⧖	▶
	Panigrahi et al. [204]	2019	■	🔍	🔓	🔑	▶	⁘	⌛	▶
Others	Shin et al. [205]	2018	■	✖	🔓	git	◀	⁘	⧖	▶
	Zhang et al. [206]	2020	■	🔍	🔓	git	▶	🪄	⧖/⌛	Skip-Gram / CNN
	Opitz et al. [207]	2022	■	🔍	🔓	git	◀	🪄	⌛	BERT

8.3.1 Highlights

In the following part, we provide an overview of each category, as shown in Table 8.1

Adaptive Embedding Transformation. Methods in this category are frequently evaluated, with most papers performing evaluations. There is a mix of benchmarks provided and not provided, indicating some variability in the adoption of standard benchmarks. Public datasets are commonly used, but there are instances where the dataset source is not mentioned. Code availability is relatively high, with many papers providing their code. There is a balance between ante-hoc and post-hoc explanation types.

Semantic Concept Alignment. Papers in this category often provide benchmarks and perform evaluations. Public datasets are typically used, enhancing reproducibility. Code availability is good, with several papers sharing their implementation. Both ante-hoc and post-hoc explanation types are represented.

Semantic Enrichment through Knowledge Integration. This category shows a trend towards providing benchmarks and performing evaluations. Public datasets are prevalent, with fewer instances of datasets not being mentioned. Code sharing is common, facilitating replication and further research. There is a slight inclination towards post-hoc explanations.

Embedding Rotation. The methods here are less consistent in providing benchmarks and evaluations. The use of public datasets and code availability is relatively lower compared to other categories. Ante-hoc explanations are more commonly found in this category.

Topic Modelling Techniques. There is a strong emphasis on providing benchmarks and performing evaluations in this category. Public datasets are widely used, and most papers share their code. This category predominantly features ante-hoc explanation methods.

Others. This miscellaneous category shows variability, with some papers providing benchmarks and evaluations, while others do not. Public dataset usage and code availability are moderate. Explanation types are balanced between ante-hoc and post-hoc.

General Trends. The majority of papers across categories tend to perform evaluations, indicating a focus on assessing the effectiveness of the proposed methods. Public datasets are commonly used, which helps in standardising comparisons and improving reproducibility. Code availability is relatively high, reflecting the community's growing emphasis on open science. Both ante-hoc and post-hoc explanation types are well-represented, showing that there is ongoing research and interest in both approaches. Regarding the embeddings discussed, all groups predominantly addresses prediction-based and non-contextual embeddings. In terms of inference methods, however, there is a greater diversity observed, encompassing approaches such as autoencoders and skip-grams.

8.3.2 Methodology Trends

As illustrated in Fig. 8.8, our survey highlights the temporal distribution of the studied works, with the earliest identified publication dating back to 2010 and a notable peak in publication volume occurring in 2019. This temporal analysis provides insight into the evolving interests and developments

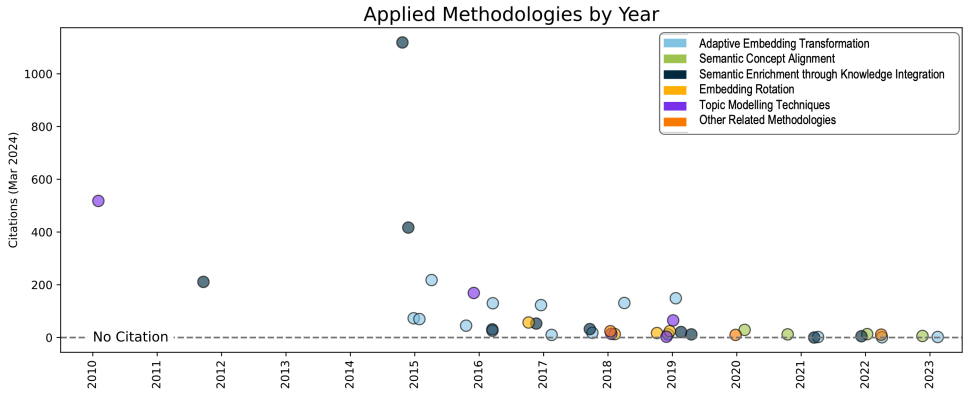


Figure 8.8: Distribution of published papers and corresponding citations by category and year (2010–2023). This graph illustrates the volume of research output and the citation impact within six categories described in Section 10.2: Category 1: Adaptive Embedding Transformation, Category 2: Semantic Concept Alignment, Category 3: Semantic Enrichment through Knowledge Integration, Category 4: Embedding Rotation, Category 5: Topic Modelling Techniques, and Category 6: Other Related Methodologies. *A deliberate horizontal jitter has been applied to distinguish overlapping data points.*

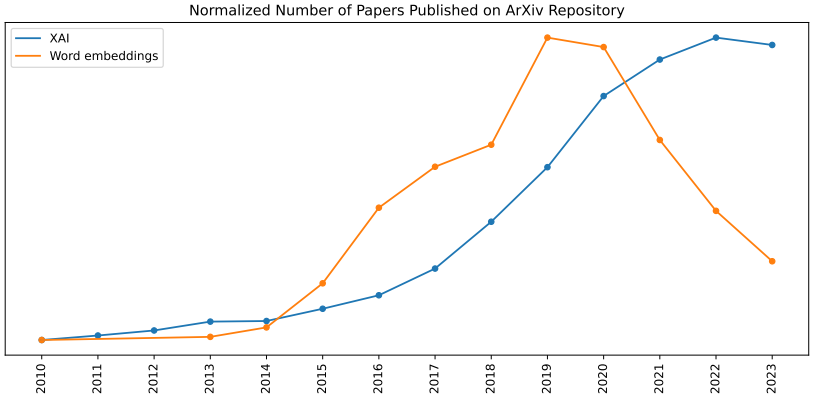


Figure 8.9: Number of published papers (normalized) regarding "XAI" and "Word Embedding" topics on ArXiv repository

within the domain. Among the various research categories, "Semantic Enrichment through Knowledge" emerges as the most cited, signifying its pivotal role and impact in advancing the field. Conversely, "Semantic Concept Alignment" is identified as the category with the least citation frequency, suggesting areas of potential growth or underexplored avenues within the research community.

Furthermore, the analysis reveals that while the methodologies of "Embedding Rotation" and "Topic Modelling" ceased to evolve past 2019, other thematic categories have continued to generate scholarly contributions up to at least 2022. This observation underscores the dynamic nature of the field, with certain research trajectories reaching maturity, while others remain vibrant and actively explored. The continuous publication within these categories beyond 2019 highlights ongoing innovations and the field's responsiveness to emerging challenges and opportunities.

Topic Trends To ensure that the observed trends are not simply reflective of the general trends in the main topics, namely XAI and Embedding, we analyzed publication data by scraping 606,814

computer science papers from the arXiv repository[1] from 2010 to 2023. We identified the number of publications for each topic using the keywords defined in Section 8.1 to filter the papers. As illustrated in Fig. 8.9, the only notable similarity with the trend of the studied works, shown in Fig. 8.8, is the peak in 2019, which is also observed in the word embedding topic. This indicates that the trends in Fig. 8.8 are likely influenced by factors specific to the studied methodologies rather than by the overall trends in XAI and Embedding publications.

8.4 Conclusion

The quest for interpretability and explainability within the field of machine learning is not merely an academic pursuit but a foundational necessity for the ethical deployment of AI technologies. Word embeddings, as a cornerstone of natural language processing, have a profound impact on a myriad of applications ranging from search engines to sentiment analysis. Despite their ubiquity and importance, our survey has unearthed a critical gap in the existing literature, i.e., to our knowledge, there has been no comprehensive survey that studies the interpretability and explainability of word embeddings in an exhaustive and structured manner.

Through careful analysis, we have classified the literature into six definitive categories, each delineating a unique methodology to achieve interpretability. Doing this classification, two pivotal insights have emerged. Firstly, there is a significant gap in the literature regarding the direct examination of the interpretability and explainability of word embeddings, including a lack of emphasis on the associated concerns, which given the influence of these models, this gap is both surprising and concerning. Secondly, and perhaps more importantly, our survey reveals that the focus of existing research is disproportionately aimed at deciphering the outputs of the 'black-box' models i.e., word embeddings. Alarmingly, the inner workings of these 'black-box' models—the very algorithms that craft the embeddings we so heavily rely upon—remain unexplained.

We argue that while interpretable embeddings are undoubtedly beneficial, offering clarity for their resulting applications, explaining the 'black-box' that generates these embeddings is equally critical. This is particularly true in an era where biases embedded within data can perpetuate and magnify societal inequities. By only addressing the interpretability of the outcomes, we neglect the roots of potential biases that may be encoded within the models themselves.

In response to these revelations, our survey stands as a call to the research community. We advocate for a paradigm shift that seeks to penetrate the surface of model outputs and to illuminate the opaque processes within the 'black-box'. This shift is not without its challenges; it requires innovation in methodology and a commitment to transparency. However, the benefits—such as ensuring fairness, enhancing trust, and enabling more robust applications—far outweigh the difficulties.

As such, we encourage researchers and practitioners alike to recognise the dual importance of interpretability and explainability in word embeddings and generating models. This dual focus is imperative for advancing a responsible and equitable AI future. We invite fellow researchers to undertake this vital journey, to explore the uncharted territories of the 'black-box', and to contribute to a body of work that will fortify the integrity and accountability of machine learning technologies.

8.4.1 Future Research Directions

The study of XAI in the context of word embeddings is still in its nascent stages, and there are numerous avenues for future research. Based on the findings and gaps identified in this survey, we propose the following research questions to guide future investigations:

[1] https://arxiv.org

Enhancing Interpretability Without Compromising Performance. *How can we enhance the interpretability of word embeddings generated by black-box models without compromising their performance in downstream NLP tasks?* One critical area of research is the development of methods that can enhance the interpretability of word embeddings generated by black-box models without compromising their performance in downstream NLP tasks. This balance is crucial to ensure that the benefits of high-performing models are not lost while making them more transparent and understandable.

Mitigating Biases in Word Embedding Models. *What methodologies can be developed to reveal and mitigate biases embedded within word embedding models to ensure fairness and equity in AI systems?* Another vital research direction is to explore methodologies for revealing and mitigating biases embedded within word embedding models. Addressing biases is essential to ensure the fairness and equity of AI systems, especially as these models are increasingly used in decision-making processes that affect people's lives.

Integrating External Structured Knowledge. *How can we effectively integrate external structured knowledge into word embeddings to enhance their semantic coherence and interpretability?* Future research should also focus on integrating external structured knowledge into word embeddings to enhance their semantic coherence and interpretability. By enriching word embeddings with external knowledge sources, researchers can improve their alignment with human understanding and make them more useful for practical applications.

Advanced Visualisation Techniques. *What are the most effective techniques for visualising the latent semantic structures within word embeddings to facilitate their analysis and interpretation by humans?* Developing advanced visualisation techniques for latent semantic structures within word embeddings is another promising research direction. Effective visualisation tools can facilitate the analysis and interpretation of complex embedding structures, making them more accessible to researchers and practitioners.

Benchmarking Interpretability and Explainability. *How can we develop and validate benchmarks for evaluating the interpretability and explainability of different word embedding models?* Lastly, there is a need for standardised benchmarks to evaluate the interpretability and explainability of different word embedding models. Developing and validating these benchmarks will ensure consistent evaluation criteria across studies and help in the systematic comparison of various approaches.

These research directions aim to bridge the gap between the technical advancements in word embeddings and the growing need for transparency and interpretability in practical applications. Addressing these questions will contribute to the development of more transparent, fair, and trustworthy AI systems.

Part V

Vector Space Model for Labor market

Chapter 9

KRAKEN: A novel semantic-based approach for keyphrases extraction

Keyphrase Extraction (KPE) is a fundamental task in Natural Language Processing (NLP), aimed at automatically identifying the most informative words and expressions that capture the essential content of a text. Within the broader field of Labour Market Intelligence (LMI), KPE techniques play a crucial role in transforming large volumes of unstructured textual data—such as online job advertisements (OJAs), curricula vitae, and training descriptions—into structured and analyzable information. By extracting key concepts and domain-relevant terms, KPE enables the identification of emerging skills, occupational trends, and evolving labour market needs, thus serving as a bridge between raw text data and structured analytical frameworks.

This chapter presents a new unsupervised approach to key phrase extraction called `KRAKEN` [214] (**K**eyphrease ext**RA**ction ma**K**ing use of **E**mbeddi**N**gs). This method leverages widely adopted NLP and distributional semantics techniques to identify representative keyphrases in textual documents without the need for labeled data. `KRAKEN` combines traditional text preprocessing steps with word embedding models to obtain dense vector representations that preserve semantic meaning. Unlike purely statistical approaches, KRAKEN explicitly exploits the relationships among words within the text by modeling their proximity and contextual similarity in the embedding space.

Beyond its methodological contribution, KRAKEN has been applied to a real-world scenario within the European project **NES (New Emerging Skills)**, aimed at identifying novel and emerging skills from 5M+ OJAs collected in five different languages across the project partner countries (UK, IT, FR, RO, and ES). The project is part of an ongoing EU grant coordinated by Eurostat and Cedefop, whose objective is to build the first EU-wide real-time labour market monitoring system by collecting and classifying online job advertisements from all 27 + 1 European countries and in 32 languages [7, 63, 55, 10, 215]. Within this framework, KRAKEN was employed as a core component of the *NES* pipeline, serving as a baseline for automatically identifying "skill sentences"—keyphrases likely to contain emerging or novel skills inferred from job ads. Through its combination of unsupervised learning, semantic embeddings, and multilingual scalability, KRAKEN represents a bridge between methodological innovation in NLP and practical applications in LMI, providing a foundation for real-time analysis of evolving skills and occupations in the European labour market.

9.1 Keyphrases Extraction Task

Keyphrases refer to a set of relevant terms that provide a high-level description of a textual document. The **Keyphrases Extraction task (KPE)** defines a range of approaches and techniques that aim to identify keyphrases. The extraction of key phrases is of significant importance in the field of natural language processing (NLP), particularly in text summarization, content recommendation or topic

modeling tasks. We provide a formal definition of KPE task:

Definition 9.1.1 (Keyphrases Extraction task (KPE))**.** Let $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ the corpus consisting of n documents, $d_i = \langle t_1, t_2, \ldots, t_m \rangle$ a single document of length m consisting of a sequence of term t_j in the order of how they appear in the document itself. A keyphrases for a document d_i is a subsequence of terms in d_i: $kp_i = \langle t_l, t_{l+1}, \ldots, t_r : \forall\, 1 \leq l, l \leq r \leq m \rangle$, so in this context a keyphrase can also consist of only a single word.

The typical keyphrase extraction pipeline typically involves two steps:

1. Keyphrases extraction: In this step, specific algorithms are used to identify and extract a set of candidate keyphrases for a text.

2. Keyphrases ranking: Following the extraction step, a rank is applied to determine the best keyphrases among those extracted. This ranking can be used for evaluation purposes or to perform a specific task.

9.2 KRAKEN: A novel approach to KPE

KRAKEN introduces a novel unsupervised approach for keyphrase extraction that leverages semantic relationships captured by word embeddings. The method begins with a standard preprocessing stage that includes tokenization, lowercasing, and the removal of stop words and punctuation. The focus is then placed on *nouns* and *adjectives*, which are selected as candidate terms since they typically encapsulate the most informative content of a text [216]. These parts of speech are thus treated as anchors for keyphrase construction.

The extraction process operates iteratively by constructing a *contextual window* around each candidate word (noun or adjective). This window includes the words immediately preceding and following the anchor term, forming potential keyphrase candidates. In this framework, each keyphrase corresponds to a localized semantic unit centered on a specific anchor word.

To determine the optimal composition of each window, KRAKEN exploits the semantic similarity encoded in a pre-trained word embedding model. Specifically, it evaluates the cosine proximity between the vector representation of the current window and those of surrounding words that can be appended to it. This proximity-based analysis guides the expansion or restriction of the window, ensuring that only semantically coherent and contextually relevant terms are included in the final keyphrase. As a result, KRAKEN can capture multi-word expressions that reflect meaningful concepts rather than relying solely on syntactic adjacency or frequency-based heuristics.

As illustrated in Figure 9.1, the implementation of the keyphrase extraction process in KRAKEN follows a two-step workflow. The first step involves the identification and extraction of candidate keyphrases from the text, based on the semantic relationships between words. The second step focuses on ranking and selecting the most representative keyphrases by applying specific evaluation metrics and comparing the results against established baselines.

Two variants of KRAKEN are proposed, differing in the similarity measure used for window construction and evaluation. The first variant employs the Pearson correlation coefficient to assess the linear dependency between word vectors, while the second relies on Cosine similarity to capture their angular proximity in the embedding space. Both approaches are systematically compared with state-of-the-art methods, as discussed in the baseline evaluation section.

Step 1: Keyphrases Extraction

In the first step of the keyphrase extraction process, two main objectives are addressed: (i) performing comprehensive preprocessing to reduce noise in the input text and ensure a clean corpus for the word embedding model, and (ii) identifying and extracting meaningful keyphrases.

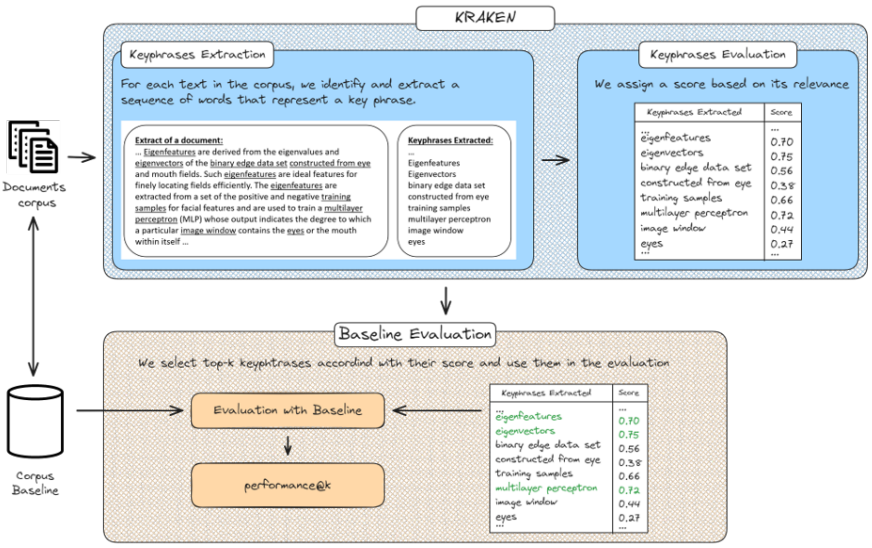


Figure 9.1: The KRAKEN workflow and keyphrases evaluation.

We apply state-of-the-art preprocessing operations to prepare the corpus. First, all text is converted to lowercase. Then, stop words, numbers, punctuation marks, accent marks, diacritics, and HTML tags are removed. Lemmatization is subsequently applied to obtain the base form of each word by removing inflectional endings. Finally, to enrich the representation capacity of the embedding model, n-grams are identified in the text, particularly unigrams, bigrams, and trigrams. These preprocessing steps contribute to improving the overall quality of the corpus and enhance the performance of the keyphrase extraction process [217, 218, 219].

After cleaning the corpus, a word embedding model is trained, which is later used to construct the context windows around potential keyphrases. Part-of-speech (POS) tagging is then applied to identify nouns and adjectives, referred to as *anchor* words, which serve as the central elements for keyphrase identification.

The algorithm iteratively builds phrases around each anchor word by analyzing the surrounding context. It first considers the words preceding the anchor and then those following it. This process is guided by a relatedness measure computed between the embedding vector of the current window and that of each candidate word. In the base case, when the window contains only the anchor word, the relatedness between the anchor and its immediate neighbors is calculated. If this value exceeds a fixed threshold, the neighboring word is added to the window. Otherwise, the construction process stops.

At each iteration, the relatedness between the embedding representation of the current window $\vec{v}_{kp_{i-1}}$ and the vector of a new word $\vec{v}_{w_i}$ is computed. Let α_i denote this relatedness measure. The construction process proceeds as follows:

1. Compute the relatedness α_i between $\vec{v}_{kp_{i-1}}$ and $\vec{v}_{w_i}$.
2. If α_i is greater than the relatedness obtained in the previous iteration α_{i-1}, proceed; otherwise, stop expanding the phrase.
3. Add w_i to the current keyphrase.
4. Repeat steps 1-3 for the next words, updating the keyphrase representation at each iteration.
5. Stop when a word decreases the relatedness or when no additional words can be added.

Thus, the algorithm incrementally constructs keyphrases by incorporating words that strengthen the semantic coherence of the window, stopping once this coherence begins to deteriorate or the phrase

reaches its intended length. The definition of relatedness α depends on the variant of KRAKEN being used: for the Pearson correlation version, α corresponds to the correlation coefficient $\rho_{\vec{v}_{kp_{i-1}},\vec{v}_{w_i}}$, while for the cosine similarity version, it represents the cosine of the angle between the two vectors.

Data: a document d
Result: the set of keyphrases extracted from the document
$kp_d \leftarrow \emptyset$;
for *word* ***in*** d **do**
  if *word is a noun* ***or*** *an adjective* **then**
    $kp_L \leftarrow$ *extract_kp(word, d)* ; // Extract phrase on the left side
    $kp_R \leftarrow$ *extract_kp(word, d)* ; // Extract phrase on the right side
    $kp_{word} \leftarrow kp_L \cup word \cup kp_R$;
    $kp_d \leftarrow kp_d \cup kp_{word}$;
  end
end
return kp_d

Algorithm 1: Extract keyphrases from a document.

Two stopping criteria govern the construction of the context windows: (i) if the initial relatedness is below a threshold t, no word is added, and the window consists only of the anchor word; (ii) if the relatedness between a new word and the current window decreases compared to the previous iteration. However, to avoid premature termination when a slightly less related word might lead to a better phrase, the algorithm allows one tolerance step: if the relatedness decreases for a single word, that word is still included, but if a subsequent decrease occurs, the process stops. This modification ensures smoother window expansion and helps capture semantically coherent phrases.

Algorithm 1 summarizes the extraction procedure: for each noun or adjective in the text, a keyphrase is iteratively constructed by extending to the left and right sides of the anchor word. The output kp_d is the complete set of extracted keyphrases for document d.

Step 2: Keyphrases Evaluation

After the keyphrase extraction step, a set of candidate phrases is obtained for each processed document. The next phase focuses on evaluating each candidate window and assigning a relevance score. Only the top-ranked phrases are retained for comparison with the baseline. The scoring process employs the same word embedding model used during the window construction stage.

Candidate phrases are categorized into two types: *short keyphrases*, composed of a single word, and *long keyphrases*, consisting of multiple words. Different evaluation measures are applied according to the type of phrase, namely the within-window score and the between-window score.

The within-window score (*WW*) is applied exclusively to multi-word keyphrases to avoid excessively long or incoherent phrases. It measures the internal semantic cohesion of the words that compose the keyphrase. Specifically, it is computed as the average relatedness between all pairs of words within the window. If this average exceeds a predefined threshold, the phrase proceeds to the next evaluation step; otherwise, it is discarded. The threshold parameter also influences the number of long keyphrases that survive the filtering process: higher thresholds result in fewer, but more coherent, phrases being retained. The within-window score for a keyphrase i is defined as follows:

$$WW_{kp_i} = \frac{1}{|kp_i^{(2)}|} \sum_{\substack{w_n, w_m \in kp_i \\ (w_n, w_m) \in kp_i^{(2)}}} \alpha(w_n, w_m) \tag{9.1}$$

where $kp_i^{(2)}$ denotes the set of all two-word combinations extracted from the words composing

kp_i, α is the relatedness measure (either cosine similarity or Pearson correlation), and w_n, w_m are two words within the same keyphrase.

The long keyphrases that pass the within-window evaluation are then combined with the single-word candidates. This merged set is evaluated using the between-window score (*BW*), which quantifies the relationship between each keyphrase and all other keyphrases extracted from the same document. The *BW* score reflects the semantic centrality of a phrase, i.e., its ability to represent the overall meaning of the text.

A higher *BW* score indicates that the keyphrase captures the most representative concepts of the document. The *BW* score for a keyphrase *i* in document *d* is defined as:

$$BW_{kp_{i,d}} = \frac{1}{|kp_d| - 1} \sum_{\substack{kp_{j,d} \in kp_d \\ kp_{i,d} \neq kp_{j,d}}} \alpha(kp_i, kp_j) \tag{9.2}$$

where kp_d is the set of all keyphrase candidates for document d, and α again denotes the similarity measure (cosine or Pearson correlation).

Data: a list of keyphrases kp_d for a document d, and a threshold th_{WW} for filtering multi-word keyphrases
Result: a ranked list of keyphrases based on the between-window score
for *kp* ***in*** kp_d **do**
  if *kp is a multi-word keyphrase* **then**
    $WW_{kp} \leftarrow$ *Compute Eq. (9.1)*;
    if $WW_{kp} \geq th_{WW}$ **then**
      $BW_{kp} \leftarrow$ *Compute Eq. (9.2)*;
    end
  else
    $BW_{kp} \leftarrow$ *Compute Eq. (9.2)*;
  end
end
`// Order keyphrases by their BW score;`
$sorted\_kp \leftarrow order(kp_d)$;
return *sorted_kp*

Algorithm 2: Calculation of scores and ranking of keyphrases.

Algorithm 2 summarizes the ranking procedure. After filtering long keyphrases based on the within-window score threshold th_{WW}, all remaining candidates are evaluated using the between-window score. The final output is an ordered list of keyphrases ranked by their *BW* score, representing their overall relevance within the document.

9.3 Baseline Evaluation

The results obtained by applying the two versions of `KRAKEN` to a set of benchmark datasets are presented and compared against several state-of-the-art approaches for keyphrase extraction.

The benchmark datasets used in this evaluation are widely adopted in the NLP community for assessing the performance of keyphrase extraction methods. Specifically, five English datasets were selected, all publicly available on GitHub[1]. Each dataset consists of a collection of documents paired with a gold standard list of keyphrases manually annotated by human experts or authors. Table 9.1

[1] Available at `https://github.com/LIAAD/KeywordExtractor-Datasets`

Table 9.1: Statistics of the benchmark datasets used for evaluation.

Dataset	# Documents	Avg. Text Length (words)	Avg. Keyphrases	Avg. Single-word KPs	Avg. Multi-word KPs
Inspec	2000	124.36	14.11	2.32	11.79
KDD	755	190.70	4.10	1.04	3.05
SemEval2010	243	8032.55	15.58	3.12	12.45
Nguyen2007	209	5121.67	12.01	3.31	8.69
WWW	1330	82.04	4.82	1.65	3.16

reports descriptive statistics for each dataset, including the number of documents, the average text length, and the average number of keyphrases per document.

The *SemEval2010* dataset [220] includes 244 scientific papers extracted from the ACM Digital Library. These papers span four major areas of computer science: distributed systems, information retrieval, distributed artificial intelligence, and social and behavioral sciences. The keyphrases were assigned by the authors or editors of the papers themselves.

The *KDD* dataset [221] contains abstracts of papers published in the ACM Conference on Knowledge Discovery and Data Mining (KDD) between 2004 and 2014, for a total of 757 documents. Each abstract is accompanied by author-assigned keywords.

Similarly, the *WWW* dataset [221] includes 1,330 abstracts from the World Wide Web Conference (WWW) over the same period, also annotated with author-provided keywords.

The *Nguyen2007* dataset [222] comprises 211 full-text conference papers. In this case, the gold standard keyphrases were manually annotated by student volunteers, each responsible for reading and labeling three papers.

Finally, the *Inspec* dataset [223] includes 2,000 abstracts from scientific journal papers in computer science, collected between 1998 and 2002. Each document includes two types of keywords: (i) *controlled keywords*, which are listed in the Inspec thesaurus and may not appear in the document, and (ii) *uncontrolled keywords*, freely chosen by the editors.

Table 9.1 summarizes the key characteristics of all datasets, showing their varying lengths and annotation styles. This diversity ensures a comprehensive evaluation of `KRAKEN` across different textual domains and annotation schemes.

Comparison with Existing Approaches Several unsupervised and weakly supervised approaches have been proposed in the literature for automatic keyphrase extraction. In this section, we briefly describe the main methods used as baselines for comparison with `KRAKEN`.

Chi et al. [224] introduced **ISKE**, a PageRank-inspired method designed to weight relationships between sentences based on the assumption that strong causal links exist between adjacent sentences. Instead of iterating over words, ISKE constructs a graph at the sentence level, where edges are weighted according to lexical overlap between sentences. This design reduces both computational and temporal complexity while preserving semantic relationships within the text.

Liu et al. [225] proposed **TopicPageRank (TPR)**, which combines topic modeling with graph-based ranking. In this approach, a co-occurrence graph is built for each document using word co-occurrence statistics within a fixed-size window. PageRank is then applied at the topic level to assign weights to words according to their importance within the discovered topics, effectively integrating topical and structural information.

Boudin [226] presented **MUL** a multipartite graph-based approach that explicitly models topical information and keyphrase candidates in the same structure. In this model, nodes represent candidate keyphrases, and edges connect keyphrases belonging to different topics. Edge weights are computed based on keyphrase co-occurrence statistics. A TextRank algorithm is then applied to this multipartite graph to rank the nodes and identify the most relevant keyphrases.

Campos et al. [227] proposed **YAKE**, a feature-based unsupervised method. For each candidate

keyphrase, a set of statistical and positional features is extracted, including term frequency, position in the document, word casing, and the number of distinct words occurring before or after the term. These features are combined into a heuristic scoring function that ranks candidate phrases according to their relevance and distinctiveness within the document.

Overall, these approaches represent diverse paradigms in unsupervised keyphrase extraction, ranging from purely graph-based models to hybrid and embedding-driven techniques.

Evaluation Metrics and Experimental Setup For the evaluation, the performance measures considered are *precision*@k, *recall*@k, and F_1-*measure*@k, where k indicates the number of top-ranked keyphrases considered. In this study, we adopt $k = 5$ and $k = 10$. The *precision* metric quantifies the accuracy of the system in identifying relevant keyphrases, while *recall* measures its ability to retrieve all the reference keyphrases present in the gold standard. The F_1-measure represents the harmonic mean between precision and recall, providing a balanced view of both accuracy and completeness. Two variants of KRAKEN are evaluated: $\texttt{KRAKEN}_{cos}$, which employs Cosine similarity for both phrase construction and ranking, and $\texttt{KRAKEN}_{pear}$, which instead uses the Pearson correlation index.

9.4 Thresholds optimization

KRAKEN relies on several parameters that significantly influence its performance, including those related to the training of the word embedding models used for window construction. The proposed architecture employs fastText [22] with the CBOW algorithm, a learning rate of 0.1, a vector size of 300, and 100 training epochs. These settings are consistent with the configurations adopted in [55, 228], where fastText models were trained on job advertisement data to identify occupations and job skills. Although the focus here differs—keyphrase extraction rather than skill identification—the tasks share linguistic and structural similarities that justify the reuse of these parameters.

Additional parameters directly affecting performance are the threshold for window construction (th_{win}) and the within-window threshold used to filter long keyphrases (th_{WW}). Each dataset was split into two subsets: a validation portion comprising 15% of the original texts was used to optimize threshold values, while the remaining 85% served for the final evaluation. To determine the optimal thresholds, a grid search was conducted by varying both th_{win} and th_{WW} from 0 to 0.9 with a step of 0.1. The evaluation metrics considered were $F_1@5$ and $F_1@10$.

The results of this grid search are reported in Table 9.2. For each dataset and value of k, the thresholds achieving the highest F_1-scores are identified. These optimal parameters were then used in the comparison with state-of-the-art approaches. The two variants of KRAKEN yield very similar results, both in terms of F_1-scores and optimal threshold configurations.

It can be observed that, except for the WWW dataset, the best value of th_{win} is low. This indicates that the algorithm tends to form multi-word keyphrases more easily. Conversely, for the WWW dataset, the optimal value of 0.9 suggests that the extracted keyphrases are predominantly single-word terms. Regarding the within-window threshold th_{WW}, low values (as in Inspec, KDD, and WWW) promote the inclusion of longer keyphrases, whereas higher thresholds (as in Nguyen2007 and SemEval2010) lead to the filtering of a greater number of multi-word candidates.

In summary, the choice of threshold values substantially affects both the composition and the ranking of the resulting keyphrases. Lower thresholds encourage the inclusion of multi-word expressions, while higher thresholds favor more selective filtering. Nonetheless, the variability of optimal thresholds across datasets highlights the importance of dataset-specific parameter tuning for maximizing performance.

The results reveal a notable relationship between the optimal values of the within-window threshold (th_{WW}) and the average number of words per document. Specifically, a direct proportional relationship emerges between these two variables. This finding is quantitatively supported by the calculation

Table 9.2: Results of the grid search with the best threshold values and their corresponding $F_1@k$.

		KRAKEN$_{cos}$			**KRAKEN**$_{pear}$		
k	**Dataset**	**Best** th_{win}	**Best** th_{WW}	$F_1@k$	**Best** th_{win}	**Best** th_{WW}	$F_1@k$
$k=5$	Inspec	0.0	0.1	**11.9**	0.1	0.1	11.9
	KDD	0.0	0.2	17.3	0.1	0.2	**17.4**
	Nguyen2007	0.0	0.9	**7.2**	0.1	0.9	7.2
	SemEval2010	0.0	0.9	**3.9**	0.1	0.9	**3.9**
	WWW	0.0	0.1	27.20	0.9	0.1	**27.5**
$k=10$	Inspec	0.1	0.1	**18.7**	0.1	0.1	18.7
	KDD	0.1	0.1	23.1	0.1	0.1	**23.7**
	Nguyen2007	0.1	0.9	9	0.1	0.9	**9.2**
	SemEval2010	0.1	0.9	**5.1**	0.1	0.9	5
	WWW	0.9	0.1	**29.3**	0.9	0.1	**29.3**

of Spearman's rank correlation coefficient. For $F_1@5$, the coefficient yielded $\rho = 0.94$ with a p-value of 0.01, indicating a strong positive correlation. This suggests that as the average length of documents increases, the optimal th_{WW} value for achieving the best performance also tends to rise. Consequently, the null hypothesis (H_0) of non-correlation can be rejected at the 95% confidence level. In the case of $F_1@10$, the correlation coefficient was $\rho = 0.86$ with a p-value of 0.057. Although ρ remains high, the slightly higher p-value suggests that the correlation, while strong, is not statistically significant at the 0.05 threshold.

From an interpretative standpoint, using a higher th_{WW} value favors the construction of more cohesive phrases composed of terms that exhibit strong semantic relatedness. This mechanism effectively filters out the noise generated by numerous non-relevant phrases, which is especially beneficial for longer documents containing a higher proportion of irrelevant content. Conversely, in shorter texts, the number of meaningful expressions is inherently limited. In such cases, adopting a lower th_{WW} prevents the premature exclusion of relevant phrases that may display weaker internal relatedness due to sparse contextual information.

The correlation analysis between the average document length and the corresponding optimal th_{WW} values thus provides quantitative support for this phenomenon. As the text length increases, the proportion of relevant terms relative to the total number of words tends to decrease, leading to higher levels of noise. In these scenarios, stricter thresholds are advantageous for isolating semantically cohesive phrases. In contrast, shorter texts benefit from more permissive thresholds that preserve potentially informative expressions.

These findings suggest that dataset characteristics—particularly the average document length—play a decisive role in determining the optimal threshold configuration. Adapting th_{WW} to the linguistic properties of the corpus can therefore enhance the overall effectiveness of keyphrase extraction.

9.5 Baseline Results

KRAKEN, in both its cosine and Pearson correlation variants, was compared against four state-of-the-art approaches described in Section 9.3: MUL, TPR, and ISKE, which are graph-based methods, and YAKE, a feature-based method. For each dataset, results are reported in terms of *precision@k* (P@k), *recall@k* (R@k), and *F1-measure@k* (F1@k), with $k = 5$ and $k = 10$.

The analysis of the results in Table 9.3 shows that, for performance@5, the Pearson correlation variant of KRAKEN achieves the best overall performance on the Inspec, KDD, and WWW datasets, while the cosine similarity version consistently ranks second. In contrast, on the Nguyen2007 and SemEval2010 datasets, KRAKEN reaches precision levels comparable to the other baselines but does

Table 9.3: Performance@5 on benchmark datasets. Best results in bold.

Dataset	Eval	MUL	TPR	ISKE	YAKE	KRAKEN$_{cos}$	KRAKEN$_{pear}$
Inspec	P@5	3.9	2.7	4.0	1.4	**13.2**	**13.2**
	R@5	4.9	3.7	5.3	2.0	**9.3**	**9.3**
	F1@5	4.1	2.9	4.2	1.5	**10.5**	**10.5**
KDD	P@5	10.1	8.1	12.0	3.1	15.0	**15.3**
	R@5	12.2	9.7	14.3	4.0	15.0	**15.2**
	F1@5	10.7	8.5	12.3	3.4	15.0	**15.3**
Nguyen2007	P@5	**14.1**	10.2	12.2	10.1	11.9	11.9
	R@5	**17.7**	12.8	15.0	12.6	7.3	7.3
	F1@5	**15.3**	11.0	13.1	10.9	8.7	8.7
SemEval2010	P@5	**8.7**	5.9	7.9	4.3	7.1	7.1
	R@5	**11.9**	8.4	11.3	6.1	3.5	3.5
	F1@5	**9.7**	6.7	9.0	4.9	4.6	4.6
WWW	P@5	12.2	9.4	12.8	4.4	24.3	**24.4**
	R@5	12.9	10.2	13.9	5.0	24.3	**24.4**
	F1@5	12.0	9.3	12.7	4.5	24.3	**24.4**

Table 9.4: Performance@10 on benchmark datasets. Best results in bold.

Dataset	Eval	MUL	TPR	ISKE	YAKE	KRAKEN$_{cos}$	KRAKEN$_{pear}$
Inspec	P@10	2.9	2.6	3.3	1.3	**16.2**	**16.2**
	R@10	7.1	6.5	8.0	3.5	**15.4**	15.3
	F1@10	3.9	3.6	4.4	1.8	**15.7**	**15.7**
KDD	P@10	7.4	7.4	9.1	3.5	21.6	**21.9**
	R@10	16.8	16.4	22.0	8.8	21.6	**21.9**
	F1@10	9.7	9.7	12.2	4.8	21.6	**21.9**
Nguyen2007	P@10	9.3	8.4	10.8	9.4	11.4	**11.5**
	R@10	23.4	20.5	**25.4**	23.5	10.4	10.5
	F1@10	13.1	11.7	14.9	**18.0**	10.8	10.8
SemEval2010	P@10	5.9	5.7	6.1	3.7	**6.6**	6.5
	R@10	**15.7**	14.9	**15.7**	10.7	5.8	5.7
	F1@10	8.4	8.0	**8.6**	5.4	6.2	6.0
WWW	P@10	8.7	8.5	10.2	3.9	**28.6**	**28.6**
	R@10	17.1	16.7	19.8	8.6	**28.6**	**28.6**
	F1@10	10.8	10.5	12.6	5.1	**28.6**	**28.6**

not surpass MUL, which performs best in those cases.

When considering performance@10 (Table 9.4), the improvements of KRAKEN become more pronounced. For Inspec, KDD, and WWW, both KRAKEN$_{cos}$ and KRAKEN$_{pear}$ substantially outperform all other baselines. For Nguyen2007 and SemEval2010, KRAKEN achieves the highest precision values, leading to corresponding improvements in F1-measure for both datasets.

In summary, for performance@5, KRAKEN$_{pear}$ emerges as the best-performing variant in three of the five datasets, with KRAKEN$_{cos}$ consistently close behind. When expanding the evaluation to performance@10, KRAKEN demonstrates superior results across most datasets, highlighting its robustness and ability to maintain high-quality keyphrase extraction performance across different textual domains.

9.6 LMI Application: Identifying Skills from Million OJAs

The New Emerging Skills (NES) project is an initiative developed by CRISP to automatically identify new skills as they emerge from real labor market demand by analyzing Online Job Advertisements (OJAs). Its main goal is to support the continuous updating of the ESCO taxonomy, ensuring that it

Table 9.5: A sample of ICT-related occupation descriptions with their unique identifier codes

ISCO code	Occupation Description
1330	ICT service managers
2511	Systems analysts
2512	Software developers
2513	Web and multimedia developers
2514	Applications programmers
2519	Software and applications developers and analysts NEC
2521	Database designers and administrators
2522	Systems administrators
2523	Computer network professionals
2529	Database and network professionals NEC
3511	ICT operations technicians
3512	ICT user support technicians
3513	Computer network and systems technicians
3514	Web technicians
3521	Broadcasting and audio-visual technicians
3522	Telecommunications engineering technicians

keeps pace with the rapid evolution of occupations and skills in the labor market. Manual updating processes are time-consuming and prone to error, making automated methods like NES essential. The system combines an unsupervised methodology for discovering new skills with a graphical interface that enables users to explore and assess the detected skills. NES has been tested on millions of job advertisements collected between 2019 and 2020 from five European countries (United Kingdom, Italy, France, Spain, and Romania), and its results have been validated through expert evaluation. The project is part of an ongoing EU-funded research initiative aimed at building the first real-time European labor market monitoring system, capable of collecting and classifying job advertisements across all 27+1 EU countries and in 32 different languages [7].

The data used for the NES project consist of job advertisements published in the UK, Spain, France, Romania, and Italy between 2018 and 2019. They were collected from the main employment platforms used in the European community. The dataset comprises approximately 1.1+ million UK vacancies, while the distribution across other countries is as follows: France (2,706,739), Spain (2,161,749), Italy (865,248), and Romania (124,606).

The advertisements were categorized into ICT and non-ICT occupations. Tab. 9.5 provides a sample of ICT occupation descriptions with their corresponding ISCO codes.

9.6.1 NES

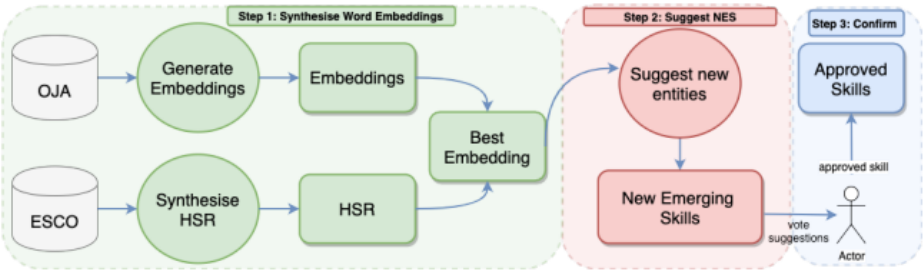


Figure 9.2: NES pipeline.

The NES system processes OJAs descriptions to identify terms that may represent new emerging skills. It leverages distributional semantics to extract semantic information and presents potential skills to users for evaluation through a graphical interface. A key objective of this project was to train word

embedding models using only the most relevant portions of job descriptions, rather than the entire text, to mitigate potential biases. To this end, KRAKEN was employed to reduce corpus dimensionality. To enhance the system's ability to distinguish between ICT-related and non-ICT skills, advertisements were categorized into ICT-ads and non-ICT-ads. Figure 9.2 illustrates the three main steps of the system pipeline.

Synthesized Word Embeddings. The initial phase involves selecting the optimal embeddings model. Three primary algorithms were evaluated with various hyperparameters:

- *Word2Vec*: Algorithm ∈ {Skip-gram, CBOW}, Hierarchical Softmax ∈ {0, 1}, vector size ∈ {5, 20, 50, 100, 300}, epochs ∈ {10, 25, 100, 200}
- *GloVe*: Vector size ∈ {5, 20, 50, 100, 300}, epochs ∈ {10, 25, 100, 200}
- *FastText*: Algorithm ∈ {Skip-gram, CBOW}, vector size ∈ {5, 20, 50, 100, 300}, epochs ∈ {10, 25, 100, 200}, learning rate ∈ {0.01, 0.05, 0.1, 0.2}

A total of 260 models were generated through grid search hyperparameter optimization, using all UK job advertisements from 2018 as the training corpus.

Model evaluation was performed using Hierarchical Semantic Similarity [55] (HSS), which measures semantic similarity between taxonomic element pairs. The selected architecture and parameters produced the vector model with the highest correlation between cosine similarity and HSS for term pairs present in both the OJA corpus and ESCO taxonomy. The optimal configuration employed FastText with CBOW architecture, vector size of 300, learning rate of 0.1, trained for 100 epochs.

Following model selection, KRAKEN was applied to generate a corpus of contextual windows for novel skill identification. To enhance embedding quality, separate models were trained using the optimal parameters for each of the eleven professional sectors across the four languages. This process yielded 44 corpora from 88 trained models, as KRAKEN requires two word embedding models: one for stop word identification and another for window construction.

The KRAKEN algorithm was adapted to construct contextual windows specifically around sentinel words rather than arbitrary nouns or adjectives. Sentinel words are terms that have higher probability of occurring in sentences containing skills. The underlying assumption is that skills tend to co-occur in textual contexts, enabling targeted window construction around these semantically rich anchor points.

Algorithm 3 presents a modified version of the original KRAKEN. The enhanced version takes as additional input the list of ESCO skills and a similarity threshold for identifying sentinel words. Cosine similarity is computed between each word in the advertisement and all ESCO skills; when the similarity exceeds the threshold, contextual windows are constructed around the identified sentinel word.

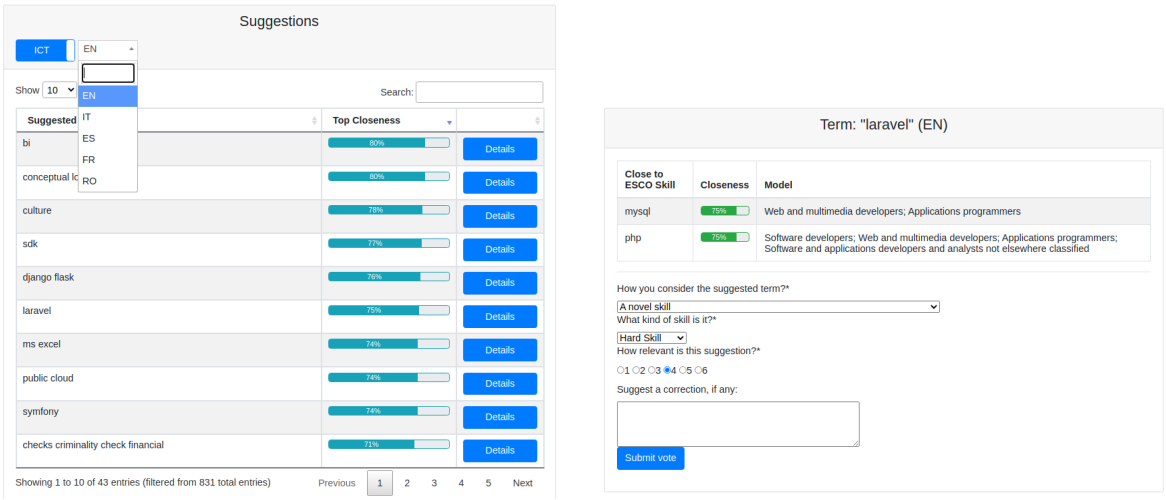
The algorithm's complexity increases due to nested loops, resulting in $O(n \cdot m)$ complexity where n represents the advertisement length and m the number of ESCO skills. In the worst-case scenario where all advertisement words are considered skills, complexity becomes $O(n^2)$. After algorithm application, the OJA collection is transformed into a set of contextualized sentences suitable for word embedding processing.

Suggest NES. The second phase focuses on extracting potential new emerging skills from the processed OJA corpus. For each ESCO taxonomy skill, we identify corpus terms with the highest cosine similarity to the target skill. A new word embedding model—using the previously optimized parameters—was trained on the KRAKEN-processed corpus to generate vectors for similarity computation.

Through cosine similarity analysis, we identified two categories of potential new skills. To ensure relevance, we filtered out low-frequency terms, retaining only those with substantial corpus occurrence. Additionally, to prioritize genuine novelty, we excluded terms exhibiting excessive semantic similarity to existing ESCO skills.

Input: *vacancy, ESCO_skills, threshold_skill, threshold_windows*
Output: The set of windows for an OJA
text ← vacancy.*split*();
vacancy_windows ← [];
for *word* ***in*** *text* **do**
  for *skill* ***in*** *ESCO_skills* **do**
    vec_word ← *get_embeddings*(word);
    vec_skill ← *get_embeddings*(skill);
    sim ← *get_cosine_similarity*(vec_word, vec_skill);
    if *sim* > *threshold_skill* **then**
      word_ind ← text.*index_of*(word);
      window_L ← *incorporate_words(word, text[:word_ind], threshold_windows)*;
      window_R ← *incorporate_words(word, text[word_ind+1:], threshold_windows)*;
      word_window ← window_L + [word] + window_R;
      vacancy_windows.*append*(word_window);
    end
  end
end
return *vacancy_windows*

Algorithm 3: *get_vacancy_windows*



(a) Language and skill type selection for suggestion review.

(b) Detailed view showing closest ESCO skills and user feedback form.

Figure 9.3: NES interface components: (a) language and skill type selection; (b) detailed skill evaluation with ESCO similarity and user feedback.

Confirm. The final stage validates identified skills through human assessment. Users first select their preferred language and specify whether to explore ICT or non-ICT skills. As shown in Figure 9.3b, each suggested term displays its highest cosine similarity (labeled "closeness") with existing ESCO skills. Evaluators assess three properties of proposed terms: (i) classification as a *new skill*, *specification*, *generalization*, *synonym*, or *irrelevant term*; (ii) categorization as *soft*, *hard*, *digital* skill, or *none*; and (iii) relevance rating on a 6-point scale (lower values indicate better suggestions). Finally, users can flag terms needing rephrasing and suggest alternative formulations.

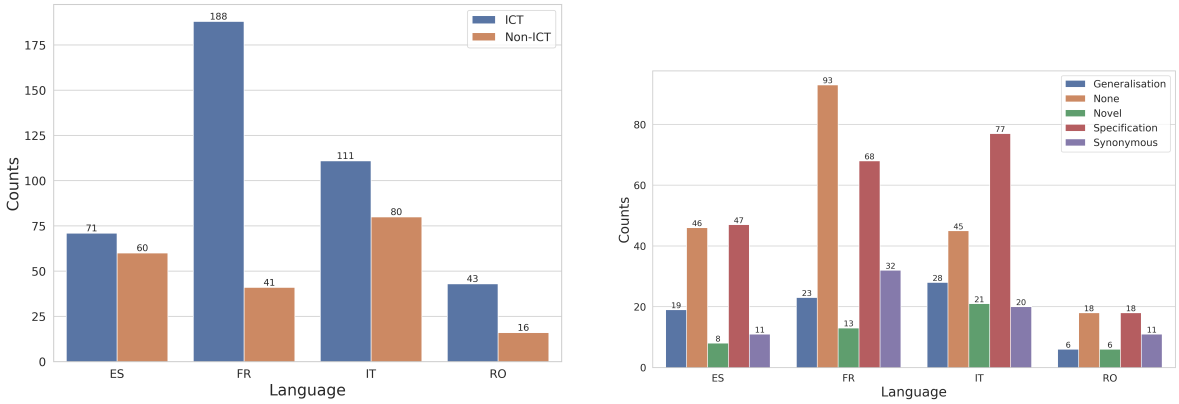
As illustrated in Figure 9.3a, after selecting from five available languages (EN, IT, ES, FR, RO) and skill type (ICT or non-ICT), the system displays corresponding suggestions. Each result shows

the closest ESCO skill matches and provides access to detailed term examination.

Figure 9.3b demonstrates the suggestion for Laravel, an open-source PHP web framework. The system identifies MySQL—a database management system supported by Laravel—and PHP, the underlying framework language, as relevant associated skills.

9.6.2 User Evaluation

To assess the effectiveness of NES, domain experts from Spain, France, Italy, and Romania evaluated the system's suggestions based on the criteria previously described.



(a) Distribution of proposed ICT and non-ICT skills across languages.

(b) Distribution of skill type evaluations (novel, specification, synonym, generalization, none).

Figure 9.4: Evaluation results: (a) distribution of ICT vs. non-ICT skills across different languages; (b) categorization of suggested skills by type.

Figure 9.4a displays the distribution of proposed novel skills across ICT and non-ICT categories for each language. The analysis reveals a predominance of ICT skills across all countries: Spain (71 out of 131, 54.2%), France (188 out of 229, 82.1%), Italy (111 out of 191, 58.1%), and Romania (43 out of 59, 72.8%).

Figure 9.4b presents the distribution of skill type evaluations, categorizing suggestions as novel, generalization, specification, synonymous, or none.

Table 9.6 presents the proportion of suggested skills deemed useful and their categorical distribution. The system demonstrated strong performance across all languages, with useful skill identification rates ranging from 56.8% to 76.4%. Specifications constituted the majority of useful suggestions, indicating NES's effectiveness in identifying specialized skill variants. These results confirm that NES successfully scales to non-UK languages while maintaining its capability to detect skill novelty.

Table 9.6: Distribution and categorization of useful skills.

Country	Accuracy	Useful Skills Distribution			
		Novel	Specification	Synonym	Generalization
Spain	85/131 (64.8%)	8 (9.4%)	47 (55.3%)	11 (12.9%)	19 (22.4%)
France	130/229 (56.8%)	7 (5.4%)	68 (52.3%)	32 (24.6%)	23 (17.7%)
Italy	146/191 (76.4%)	21 (14.4%)	77 (52.7%)	20 (13.7%)	28 (19.2%)
Romania	41/59 (69.5%)	6 (14.6%)	18 (43.9%)	11 (26.9%)	6 (14.6%)

Figure 9.5 displays the results of skill categorization evaluation (hard, digital, soft, or none). Notable patterns emerge across countries: France exhibited a high proportion of hard skills (66.8%), while Spain showed substantial digital skill identification (46.6%).

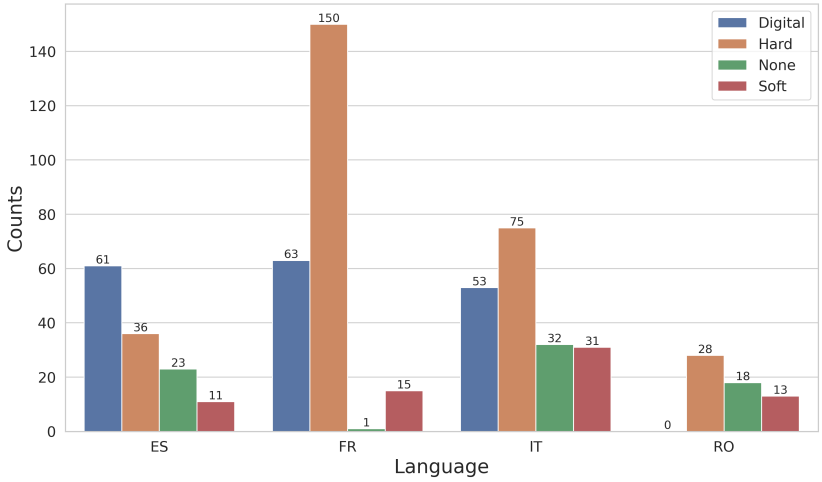


Figure 9.5: Distribution of skill category evaluations (digital, hard, soft, none).

Table 9.7 presents the relevance assessment results using a 6-point scale (1: strongly relevant, 6: not relevant). The analysis includes counts, mean scores, standard deviations, and percentile distributions across skill types and countries. Notably, novel skills in Italy received an average score of 1.9, indicating the system's effectiveness in identifying relevant new skills. Similarly, synonymous terms in Romania achieved the highest relevance (mean: 1.3), demonstrating precise semantic matching capabilities.

Table 9.7: Relevance evaluation of suggested skills (1: strongly relevant, 6: not relevant).

Skill Type	**Lang**	**Count**	**Mean**	**Std**	**Percentiles**		
					25%	**50%**	**75%**
Generalisation	ES	19	3.1	1.6	2	3	4
	FR	23	3.5	1.0	2.5	4	4
	IT	28	2.4	0.8	2	2	2
	RO	6	2.3	1.9	1.3	2	2
None	ES	46	5.6	1.1	6	6	6
	FR	99	6.0	0.3	6	6	6
	IT	45	3.9	0.5	4	4	4
	RO	18	1.6	0.7	1	1.5	2
Novel	ES	8	4.5	1.8	3.8	5	6
	FR	7	5.6	1.1	6	6	6
	IT	21	1.9	0.4	2	2	2
	RO	6	2.5	2.0	1	2	3
Specification	ES	47	2.5	1.2	2	2	3
	FR	68	4.4	1.2	4	5	5
	IT	77	2.1	0.5	2	2	2
	RO	18	2.3	1.0	2	2	2.8
Synonymous	ES	11	2.2	1.7	1	2	2
	FR	32	1.9	1.3	1	1.5	2
	IT	20	2.1	0.7	2	2	2
	RO	11	1.3	0.5	1	1	1.5

9.7 Conclusion

This work presented KRAKEN, a novel domain-independent approach for keyphrase extraction that leverages word embeddings to identify meaningful phrases by analyzing word relationships within texts. We developed two versions—**KRAKEN**$_{cos}$ and **KRAKEN**$_{pear}$—utilizing cosine similarity and Pearson correlation, respectively, to evaluate keyphrase relevance. A key innovation of KRAKEN is the separation of extraction and ranking phases, enabling modular application where phrases can be identified via window-based techniques and subsequently weighted using correlation measures.

Evaluation across benchmark datasets (Inspec, KDD, Nguyen2007, SemEval2010, WWW) demonstrated that both versions achieve competitive performance, with KRAKEN attaining top results in Precision@10 across all datasets and outperforming feature-based methods like Yake while maintaining strong performance against graph-based approaches. The method incorporates a two-step metric assessing intra-phrase cohesion and inter-phrase semantic alignment, with optimization revealing a correlation between optimal thresholds and text length.

Beyond theoretical contributions, KRAKEN was applied practically in the NES system for identifying emerging skills from 6+ million online job ads across five countries. User evaluations confirmed NES's effectiveness, with accuracy rates of 64.8% (ES), 56.8% (FR), 76.1% (IT), and 69.5% (RO), and relevance scores averaging 1.3-2.3 on a 6-point scale. These results underscore KRAKEN's robustness in both keyphrase extraction and real-world skill identification applications.

Chapter 10

JobSet: Synthetic Job Advertisements Dataset for Labour Market Intelligence

This chapter introduces a significant contribution to addressing one of the fundamental challenges in computational labor market analysis: the scarcity of high-quality annotated data for training machine learning models. The exponential growth of Online Job Advertisements (OJAs) over the past decade has created unprecedented opportunities for real-time labor market observation, skill trend prediction, and evidence-based policy making. Particularly relevant in the context of the European Union's declaration of 2023 as the Year of Skills, the need for accurate skill intelligence has never been more critical. However, the potential of machine learning approaches has been consistently hampered by the limited availability of annotated OJA datasets, which require extensive manual effort and often fail to represent the true diversity of the online job market.

To bridge this critical gap, this thesis presents JobGen [229], an innovative framework that leverages Large Language Models (LLMs) to generate synthetic yet realistic OJAs. Our approach fundamentally addresses the data scarcity problem by combining real OJAs collected from European projects with the structured knowledge of the ESCO taxonomy, ensuring both distributional accuracy and semantic coherence. The core contribution of this work is JobSet, a comprehensive dataset of synthetic OJAs that provides researchers and practitioners with a valuable resource for developing and evaluating machine learning algorithms in labor market applications.

The development of JobSetrepresents a strategic advancement in this thesis, complementing the methodological contributions of previous chapters by addressing the fundamental data requirements that enable robust skill extraction, job classification, and market analysis. By making this dataset openly available to the research community, we not only facilitate immediate applications in skill intelligence but also establish a foundation for reproducible research and comparative evaluation of computational methods in labor market analysis.

This work is partially supported by the research activity of a grant entitled *"PILLARS — Pathways to Inclusive Labour Markets"* - under the call H-2020 TRANSFORMATIONS 18-2020 "Technological transformations, skills and globalization - future challenges for shared prosperity", grant agreement NUMBER 101004703 which aims at using Labour Market Data across Europe to predict future jobs[1].

10.1 Introduction

Over the past years, the online labour market has experienced substantial growth, offering unprecedented opportunities for gaining insights into workforce demand and supply using Machine Learning (ML) to understand labour market dynamics and support policy making. Since the early 2010s, the problem of collecting and analysing OJAs has become key for companies and public institutions, as

[1] https://www.h2020-pillars.eu/

in the case of Cedefop EU Agency and Eurostat, given the ability of OJAs to observe labour market phenomena in real-time, supporting policy and decision-making (see, e.g. [230, 53]). In 2019, Cedefop and Eurostat joined forces to build up the European Web Intelligence Hub (WIH), which has been collecting OJAs from 700+ sources over 27+1 EU Countries, handling 450M+ unique OJAs in 28 languages, aiming to put OJAs into official statistics. To this end, the WIH [2] uses ensemble learning techniques to classify job ads over standard taxonomies and extract skills. All these initiatives shed light on the practical significance of collecting and sharing annotated OJAs representative of the real online labour market demand regarding occupation distributions and skills, acting as a benchmark to support advances in online labour market analytics. This includes, among others, extracting insights from job advertisements [127, 10], assessing the demand for skills and occupations [130, 55], and enhancing the accuracy of job seeker-opportunity matches [144, 43]. However, those tasks are bound to the presence of OJA datasets labelled to train the ML models and evaluate the results. While some datasets exist in previous literature, only a few have been made available, often restricted in size and coverage of skills and occupations. Furthermore, none of them has provided annotation guidelines, thus obscuring the interpretation of competencies [146]. Finally, most of the available datasets required manual annotation [146], which is labour-intensive, time-consuming, and error-prone. This manual process often limits dataset size and coverage, leading to bottlenecks in maintaining and updating resources. These limitations can be overcome by automatically generating synthetic data [231].

Recently, Large Language Models (LLMs) have shown promise in generating, with only a few examples, word distributions that closely mirror those found in real-world datasets if enhanced with grounding, filtering, and taxonomy-based generation strategies [25, 139, 140]. In the labour market, some recent approaches have used LLMs to construct synthetic datasets of OJAs [133, 132, 147]. However, generating data that are semantically correlated with the real ones and exhibit sufficient intra-dataset diversity remains challenging [136, 135, 134]. Moreover, previous datasets consist of short job descriptions featuring few skills. Those differences from real data can significantly impair the performance of downstream tasks.

This work presents `JobSet`, a novel dataset of OJAs, and `JobGen`, the framework developed to generate it using LLMs enhanced with taxonomic information and real-world examples and distributions. `JobSet` is released under the *Creative Commons Attribution 4.0 International* license.

Contribution. The contribution of this work is two-fold:

1. We introduce `JobGen`, a novel methodology designed for automatic data generation via knowledge-enhanced LLMs. `JobGen` is characterised by an iterative process encompassing generation and fitness assessment. The generation is based on real-world data distributions, ensuring it accurately reflects the complexities and nuances of real-life data scenarios. In contrast, the fitness assessment ensures the intra-dataset's diversity and the generated data's quality.
2. We release `JobSet`, a dataset of synthetic OJAs made available for the training and evaluation of ML algorithms designed for tasks in the domain of the labour market.

10.2 Introducing `JobGen`

The framework `JobGen` consists of four primary components, Fig. 10.1 shows an overview of the workflow: ***Step 1: Occupation-Skill Combination Generation.*** We start by extracting all occupations from the ESCO taxonomy. Exclusions apply to occupations not typically recruited through online job advertisements, such as politicians and armed forces personnel. Each occupation's real-world frequency drives the content generation, ensuring a balanced representation adjusted by a logarithmic scale to represent less common roles fairly. Additionally, we analyse the distribution of skills from the

[2] `https://www.cedefop.europa.eu/en/tools/skills-online-vacancies`

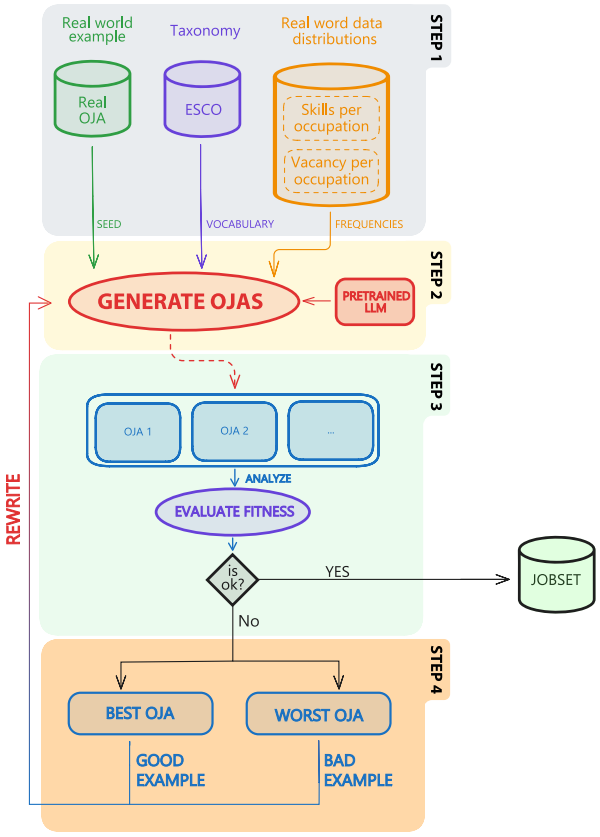


Figure 10.1: Synthesising online job advertisements guided by ESCO taxonomy and real-world data.

WIH dataset[3] to form occupation-skill pairs. As we aim to generate advertisements faithful to real-world OJAs, the number of skills for each generation is based on a Poisson sampling, where lambda is equal to the mean number of skills presented in the real-world data for the occupation. In contrast, the actual skills are sampled using the real frequencies of the skills in the occupation as weights. While our model strives to mirror real-world data, some occupations naturally exhibit fewer associated skills. To address this, we supplement these pairings with additional skills derived from ESCO, positioned at the median distribution level for required skills and the first quartile for optional skills.

Step 2: Synthetic OJA Generation. We leverage LLaMa 3 70B [232] for its balance of output quality and efficient inference, essential for generating diverse synthetic online job advertisements. LLaMa's open-access nature further guarantees the reproducibility of our work. The initial synthetic batch for each occupation outlined in the ESCO taxonomy is the foundation for subsequent refinement. In the prompt design, we included specific instructions to ensure the model generates anonymised information, avoiding including specific company names or identifiable data. The generation process incorporates the following information in the prompt: the occupation label, a brief role description, and a comprehensive list of required skills. To provide a contextual benchmark, we also include in the prompt an example of a real-world online job advertisement from the WIH NLP dataset to offer a tangible example of the real-world structure we aim to emulate in our synthetic generation.

Step 3: Fitness Evaluation. Each synthetic advertisement undergoes evaluation using the English version of the BGE-large [233] embedding model, measuring the cosine similarity matrix of all pairs of advertisements. The process will end if the average pairwise cosine similarity across the matrix is below a pre-defined threshold. This threshold is set to balance diversity and semantic coherence. It

[3]Source of OJAs: The EUROSTAT Web Intelligence Hub, publicly available upon NDA. The 2M vacancies collected in the UK in 2023 were used.

was chosen empirically and finally set to 0.85. Otherwise, the system iterates by rephrasing lower-performing advertisements, which are too similar to other generations on average (in terms of cosine similarity between their embeddings).

Step 4: Iterative Process and Convergence. Those samples are generated again, using the instances that are the furthest and closest in terms of cosine similarity as examples and varying the skills' labels to encourage diversity while maintaining relevance to the original context. The examples instruct the LLM on what to imitate and avoid. While rewriting, we use alternative skill labels from the ESCO taxonomy to vary our prompts and enhance diversity. In ESCO, alternative labels are other terms or synonyms commonly used to refer to the same skill. The rewriting prompt is carefully constructed to maintain the role's essential skills and characteristics while encouraging stylistic and structural diversity. We update our similarity assessments by focusing on the embedding and calculating similarities only for the newly rewritten descriptions during each iteration. This approach allows for rapid convergence while maintaining a comprehensive view of the dataset's diversity. We also track metadata, such as the number of rewrites per description and the time of each refinement.

The Algo. 4 depicts the steps described above. $Pois(\lambda)$ represents the Poisson distribution with parameter λ calculated as the mean of the skill frequencies for each occupation, and $sim(o_i, o_j)$ represents the cosine similarity between the embeddings o_i and o_j obtained using the BGE-large English model.

Input: ESCO Occupation, Real OJAs, threshold
Output: JobSet
$JobSet \leftarrow \emptyset$
for $occ \in ESCO\ Occupation$ **do**
  `// Step 1: Occupation-Skill Combination Generation`
  $ojas_{occ} \leftarrow$ Real OJAs associated to occ
  $skill_{occ} \leftarrow$ occ's skills distribution on $ojas_{occ}$
  $\lambda \leftarrow \text{mean}(skill_{occ})$
  $num\_skill \sim \text{Pois}(\lambda)$
  $W \leftarrow \{\text{freq}(ojas_{occ}, s) \mid s \in skill_{occ}\}$
  $selected\_skills \leftarrow \text{sample}(skill_{occ}, num\_skill, W)$
  `// Step 2: Synthetic OJA Generation`
  $num\_oja \leftarrow \lfloor \log(\text{freq}(\text{Real OJAs}, occ)) \rfloor$
  $synth\_ojas \leftarrow \text{LLM}(occ, num\_oja, selected\_skills)$
  `// Step 3: Fitness Evaluation and Refinement`
  $sim\_ojas \leftarrow \{\text{sim}(o_i, o_j) \mid o_i, o_j \in synth\_ojas \wedge i \neq j\}$
  while $mean(sim\_ojas) > threshold$ **do**
    for $o_i \in synth\_ojas$ **do**
      $o_{best} \leftarrow \underset{o_j \in synth\_ojas \setminus \{o_i\}}{\text{argmin}} \frac{\sum_{o_k \in synth\_ojas \setminus \{o_j\}} \text{sim}(o_j, o_k)}{|synth\_ojas| - 1}$
      $o_{worst} \leftarrow \underset{o_j \in synth\_ojas \setminus \{o_i\}}{\text{argmax}} \frac{\sum_{o_k \in synth\_ojas \setminus \{o_j\}} \text{sim}(o_j, o_k)}{|synth\_ojas| - 1}$
      `// Generate new oja`
      $new\_oja \leftarrow \text{LLM}(occ, o_{best}, o_{worst}, selected\_skills)$
      $synth\_ojas \leftarrow synth\_ojas \setminus \{o_{worst}\}$
      $synth\_ojas \leftarrow synth\_ojas \cup \{new\_oja\}$
    end
    $sim\_ojas \leftarrow \{\text{sim}(o_i, o_j) \mid o_i, o_j \in synth\_ojas \wedge i \neq j\}$
  end
  $JobSet \leftarrow JobSet \cup synth\_ojas$
end
return $JobSet$

Algorithm 4: `JobGen`

This structured approach generates 15,496 unique job descriptions with associated skills and

Table 10.1: Qualitative example of a synthetic job advertisement from JobSet.

Occupation: Software Analyst

Skills set:

- Use technical drawing software
- CSS
- Computer programming
- Analyse software specifications
- Java
- SQL
- Object-oriented modelling
- ICT system programming
- Project management

Verbatim skill – Rephrased skill

Role: Software Analyst
Location: Hybrid - London/Remote based (2 days a week in the office)
Salary: GBP40,000 - GBP50,000pa
We are seeking a highly skilled Software Analyst to join our team to act as an interface between software users and the software development team. This role involves eliciting and prioritising user requirements, producing and documenting software specifications, testing their application, and reviewing them during software development. You will be an expert in technical drawing software, object-oriented modelling and be proficient in Java, CSS, ICT system programming, and computer programming.
You will work closely with stakeholders to understand their requirements and collaborate with the development team to ensure software specifications meet their needs. Your technical expertise will enable you to write SQL queries to validate data and troubleshoot ICT system programming issues.
Your skills in project management will ensure the delivery of software releases on time, within budget, and to the required quality. You will also participate in testing and reviewing software applications, providing insight and recommendations to improve software performance.

ESCO occupational classes. We used an NVIDIA A100 80GB PCIe GPU to generate the data. The total computational time for the final generation was approximately 80 hours.

10.2.1 JobSet Analysis

Dataset Composition. Each instance in the dataset represents a synthetic Online Job Advertisement consisting of synthetic text and the corresponding ESCO occupation and skills. The dataset is designed to mimic real-world job advertisements, capturing the diversity and specifics of different job roles and their associated skills. Every instance is an individual job advertisement. The dataset contains no information missing from individual instances. Each synthetic job advertisement is generated with all the necessary components. There are no explicit links or relationships between data instances. The dataset does not come with predefined or recommended data splits such as training, development/validation, and testing. Researchers and practitioners can create splits based on specific needs and experimental setups.

Dataset Statistics. Starting with 396 ESCO IV digit-level occupations, we generated 15,469 unique job postings reflecting real-world online frequency distributions. Our balancing approach achieved this, ensuring each occupation is adequately represented with an average of 39 generations per occupation class and ten average skills per generation, totalling 8,374 unique skills. JobSet comprises 4.2 million tokens across all job ads, averaging 214.8 words and 264 tokens per job advertisement. Tab. 10.1 presents an example illustrating the dataset's contents. A comparative analysis with existing

datasets is illustrated in Tab. 10.2: SKILLSPAN-M(ATCH) [146], containing over 14.5K job posting sentences scraped from various sources; DECORTE [132] is synthetically generated from ESCO using GPT-3.5-turbo; and SKILLSKAPE [147], a generated dataset containing synthetic job posting sentences generate using GPT-3.5 and labelled with ESCO skills.

Distribution. Our dataset is accessible exclusively for research purposes and can be reached using its specific digital object identifier (DOI) on Zenodo[4] [234]. The code for generating and evaluating `JobSet` is available on GitHub [5]. The dataset is distributed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license.

Table 10.2: `JobSet`' comparative statistics. Average # skills and words refer to the average per instance.

Dataset	Split	Avg. # Skills	Avg. # Words	# Samples
SKILLSPAN-M [146]	Dev.	2.0	15.0	178
	Test	1.9	16.3	751
DECORTE [132]	Train	1.0	15.7	5,120
SKILLSKAPE [147]	Train	2.6	28.2	6,352
	Dev.	2.1	27.8	1,316
	Test	2.6	28.1	1,272
`JobSet` (ours)	Total	10.0	214.75	15,469

10.3 Experiments

This section presents the set of metrics employed to evaluate the effectiveness of `JobGen` . We aim to confirm the robustness and utility of the generated outputs through two evaluation dimensions: (i) ensure that the advertisements generated are high quality and implicitly evaluated through automated assessment metrics, and (ii) explicitly evaluate baseline models performance in downstream tasks.

Table 10.3: Intrinsic quality metrics, expanded from [147].

	Perplexity	Explicitness (%)
SKILLSPAN-M	178.2	5.0
DECORTE	65.1	22.4
SKILLSKAPE	44.3	6.9
`JobSet` (ours)	**26.7**	**69.9**

Intrinsic Quality Assessment. Drawing on benchmarks from previous research [147], we use two key metrics to evaluate our dataset. **Perplexity**: This metric assesses how realistically a language model predicts text. We employ GPT-2, using a sliding window technique to handle texts longer than 1024 tokens. Lower perplexity scores indicate more natural and predictable text. While we could use a more recent model for calculating perplexity, we chose to use the same one as in previous work to maintain consistency, as it does not significantly affect the outcome but ensures comparability across

[4] `https://doi.org/10.5281/zenodo.11454052`
[5] `https://github.com/Crisp-Unimib/JobGen`

all models. **Explicitness**: Measured through string-matching, this metric counts the occurrences of exact skill phrases in the texts, i.e. the ones that are expressed verbatim. Balancing exact and rephrased skill mentions ensures our dataset can accurately train models to infer skills from varied contexts. Tab. 10.3 shows a comparative analysis of these metrics with previous works.

Extrinsic Performance on Downstream Tasks. We now detail the evaluation of our dataset's effectiveness in its applications. Testing the dataset on these tasks ensures that it meets the necessary criteria for reliability and functionality in practical scenarios.

Table 10.4: Performance (%) comparison of AI models downstream task across different evaluation metrics. The precision, recall, and F1 score are presented as weighted-aggregated values, where the weights correspond to the numerosity of the classes.

(a) Occupation Classification.

Downstream Task	Occupation Classification (I Digit)				Occupation Classification (IV Digit)			
Metric	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
all-MiniLM-L6-v2	87.88	87.88	87.88	87.84	93.37	93.68	93.37	93.00
all-mpnet-base-v2	88.59	88.59	88.59	88.66	93.15	93.26	93.15	92.70
gte-large-en-v1.5	95.99	96.00	95.99	95.99	97.06	97.16	97.06	96.91
mxbai-embed-large-v1	94.67	94.74	94.67	94.66	97.16	97.38	97.16	97.08
bge-large-en-v1.5	88.69	88.69	88.69	88.65	94.76	95.15	94.76	94.59
Models Avg	91.16	91.18	91.16	91.16	95.10	95.33	95.10	94.86

(b) Skill Extraction.

Downstream Task	Skill Extraction		
Metric	Precision	Recall	F1
all-MiniLM-L6-v2	29.42	11.60	15.15
all-mpnet-base-v2	27.35	8.63	12.12
gte-large-en-v1.5	57.49	41.99	46.66
mxbai-embed-large-v1	50.62	32.97	37.94
bge-large-en-v1.5	19.55	4.64	6.41
Models Avg	36.87	19.97	23.66

Setup. We employ several embedding models and analyse the experimental results. The models are chosen due to their high performance on the popular MTEB [235] leaderboard and are all-MiniLM-L6-v2[6], all-mpnet-base-v2[7], gte-large-en-v1.5 [236], mxbai-embed-large-v1 [237, 238] and bge-large-en-v1.5 [233]. We concatenate a simple Multi-Layer Perceptron classifier to the embedding models. This classifier is trained over 20 epochs, with the input layer size matching the one of the embedding. Two hidden layers, each consisting of 128 units, are followed by an output layer corresponding to the number of classes per occupation level or number of skills tested.

Classification of Occupations. This task involves categorising job advertisements into ESCO occupational groups based on the job vacancy provided. The accuracy and precision with which our dataset facilitates this classification reflects its utility in effectively representing job market information.

Skill Extraction. Accurately extracting and identifying skills from job advertisements is critical for matching potential candidates to job requirements and training more advanced job recommendation

[6]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[7]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

systems. This task measures how well the dataset supports extracting explicit and implied skills from text.

10.4 Discussion

Ethical Considerations. The dataset does not contain data that might be considered confidential since it comprises entirely synthetic job advertisements generated using LLMs and does not include real data from individuals or any non-public communications. Additionally, all data used in the generation process from the ESCO taxonomy and any supplemental sources are anonymised and do not contain personally identifiable information or confidential content. The dataset does not contain data that might be viewed as offensive, insulting, threatening, or anxiety-causing. Since the dataset comprises entirely synthetic data generated by computational methods, which does not involve human subjects or personal data, it did not necessitate an ethical review process. While the JobSet dataset offers numerous advantages for various labour market analytics tasks, it is unsuitable for certain applications. Future users should avoid using the dataset in automated hiring systems, which could lead to unfair treatment or biased outcomes. The dataset's synthetic nature makes it unsuitable for critical employment decisions without significant validation.

`JobSet` Compared to Previous Works. The comparative analysis of `JobSet` in Tab. 10.3 shows how our framework performs against earlier works. From this, we can note that `JobSet` includes a larger number of instances, longer sentences, and a more comprehensive range of skills per job posting. `JobSet` is constructed to be highly representative of the labour market's demands and complexities, surpassing the scope of previous datasets. For instance, our dataset captures the breadth of skills required for various occupations and reflects the depth and specificity of job descriptions necessary for effective job matching and skills development.

Moreover, as exhibited in Tab. 10.3, `JobSet` has lower perplexity than competing datasets, indicating that the language models find it easier to predict the sequence of words in our dataset. This suggests that the text in `JobSet` is more coherent and follows more predictable patterns, which is beneficial for training models that process job descriptions.

Regarding the explicitness of skill mentions, `JobSet` is closer to 50% than other datasets, which provides a clear signal of required skills while still requiring the model to perform some level of inference. This balance is important because, in real-world applications, detecting explicitly mentioned skills and those implied by context is crucial for effective job matching and skill gap analysis.

Performance Baseline on Downstream Tasks. The performance of `JobSet` on downstream tasks, including metrics such as accuracy, weighted precision, weighted recall, and weighted F1 score, is summarised in Tab. 10.4.

In occupation classification (Tab. 10.4a), the performance at the I-digit is worse than at the IV-digit level despite the fewer classes in the former. This can be attributed to the fact that I-digit classes represent a broader definition and encompass various occupations grouped by similar types of work, skill levels, and fields of activity, making it more difficult for the algorithm to classify them. For instance, the first group comprises managers, the third technicians, and the fourth clerical support workers. So we can have a *bank manager* in group one, a *bank account manager* in group 3 and a *bank teller* in group 4. In contrast, the IV-digit classification defines specific occupations with greater precision and detail.

The results for skill extraction (Tab. 10.4b) within our dataset highlight the challenging nature of the task, primarily due to the considerable variety in skills we have incorporated — totalling 8,374 distinct classes. This vast array of skills introduces a complex challenge for any classification model, explaining the more subdued performance in this area. In evaluating skill extraction, we mainly focus

on the recall metric rather than accuracy since we aim to assess the model's ability to identify and retrieve relevant skills from the job advertisements. This task falls within the Extreme Multi-label Classification (XMLC) domain and inherently presents challenges due to the large label space. This has been highlighted in various works citations [239, 146, 130, 131], where the issue of skill classification with hundreds or thousands of possible classes is addressed. The high number of classes, many of which are semantically similar or ambiguous, significantly increases the complexity of the task.

Maintenance. The dataset is designed to be robust and fully operational upon release; thus, scheduled maintenance is not generally required. It is intended to function effectively without the need for regular updates. However, significant structural changes to the ESCO framework may necessitate the generation of an updated dataset version. In such cases, the new dataset version will be recreated and made available. Older versions of the JobSet dataset will continue to be supported, hosted, and maintained through Zenodo, which provides a comprehensive version history feature. This ensures that users can access previous versions of the dataset if needed. Researchers can extend, augment, build on, or contribute to the dataset. Interested contributors can contact us with their proposed additions or modifications. Once the contribution is verified, we will incorporate the new data into the existing dataset and submit a new version, ensuring all users can access the most updated version.

10.5 Conclusion

This work introduced `JobSet`, a robust benchmark of synthetic online job advertisements. By leveraging advanced LLMs and the ESCO taxonomy, `JobSet` enhances machine learning evaluation in labour market analytics. Our framework, `JobGen`, ensures the generation of diverse and semantically aligned job ads, significantly improving the training and evaluation of models for job matching and skill assessment tasks. `JobSet` shows the effectiveness of synthetic data in addressing the challenges of manual dataset annotation and limited data availability.

Limitations. Although we aim to mimic real-world data, there may be nuances and contextual elements specific to actual job markets that the synthetic data fails to capture. As job markets evolve, the dataset will require continual updates to remain relevant. This ongoing maintenance demands resources and could become a scalability issue, especially as more languages and occupational categories are added.

Future Work. We aim to extend our dataset generation to encompass all 27+1 EU languages, enhancing our benchmark linguistic diversity and cultural relevance. Additionally, we plan to conduct expert studies involving International Country Experts to rigorously assess our dataset's quality. This will ensure that it accurately reflects the nuances and requirements of different job markets. We also intend to incorporate socioeconomic factors, enabling more targeted job advertisement generation.

Impact. By providing a high-quality, diverse dataset of synthetic job advertisements, this work can significantly improve the training and evaluation of machine learning models specifically designed for the job market. These models can benefit from a larger, more varied corpus that reflects real-world complexities, leading to more accurate and generalisable AI systems.

Resource Availability Statement. `JobSet` can be accessed via Zenodo[8] under the CC BY 4.0 license, without any IP-based or other restrictions. At the same time, the accompanying code used for generating the benchmark, evaluation scripts and evaluation results are available on GitHub[9]. Access to the WIH NLP dataset is granted for research activities by signing an NDA with Eurostat.

[8] `https://doi.org/10.5281/zenodo.11454052`
[9] `https://github.com/Crisp-Unimib/JobGen`

Chapter 11

VEUCTOR: Training and Selecting Best Vector Space Models from Online Job Ads for European Countries

Over the last decade, word embeddings have enabled machines to represent words and sentences as vectors, enabling researchers to reason on text for tasks like semantic similarity, contextual understanding, machine translation, etc. However, the synthesis of embeddings involves domain-specific parameters that affect semantic accuracy and contextual relevance, often leading to unpredictable biases and inconsistent comparisons. This issue is particularly relevant in labor market analysis, where different embeddings yield varying results, making the selection of the most appropriate model a key element.

This work addresses these challenges by (i) proposing a methodology to train, select, and align vector space models for a target taxonomy, ensuring comparability across dimensions and languages; (ii) applying this approach to 4.5 million job ads in 28 languages, aligning country-specific embeddings using the ESCO taxonomy; (iii) generating 3,000+ models over 142 machine days, making the best-performing ones publicly available via VEUCTOR[1]; and (iv) showing how model choice significantly impacts labor market analysis, revealing substantial variations in occupational skill bundles across embeddings.

This work is partially supported within the research activity of a grant entitled *"PILLARS — Pathways to Inclusive Labour Markets"* - under the call H-2020 TRANSFORMATIONS 18-2020 "Technological transformations, skills, and globalization - future challenges for shared prosperity", grant agreement NUMBER 101004703 — PILLARS[2].

11.1 Introduction and Motivation

As labor markets rapidly evolve, harnessing the power of data to decode workforce skills has never been more crucial. The European Skills Agenda[3] highlights the strategic importance of skills in driving economic competitiveness, yet measuring and identifying them remains a daunting task. In 2025, the European Union published the "Union of Skills",[4] to clarify that investing in people's skills is key for Europe's economic competitiveness, resilience, and social cohesion, thus suggesting the development of a comprehensive strategy addressing skills shortages, gaps, and mismatches, ensuring that individuals and businesses have the necessary skills for success in the evolving global economy.

[1] Under final review on Information Sciences Journal

[2] https://www.h2020-pillars.eu/

[3] https://ec.europa.eu/social/main.jsp?catId=1223&langId=en

[4] COM(2025) 90 final. https://ec.europa.eu/commission/presscorner/detail/en/ip_25_657

Notably, the Commission plans to consolidate all labour-related data —including labour shortages and surpluses reports and skills online job ads analysis tools — from agencies such as Eurostat, Cedefop, Eurofound, and the European Labour Authority into a unified data lakehouse. This, in turn, would enable real-time intelligence and skills forecasting for the Union to support policy and decision-making.

Though these initiatives highlight the importance of skills intelligence in analyzing labor market dynamics, the concept of "skill" remains ambiguous and often subject to misinterpretation, as extracting, recognizing, and analyzing them present two key challenges: The first is the development of a valid taxonomy for skill classification. The second is the identification and extraction of skills from available sources. O*NET has addressed the first challenge in the US and ESCO in Europe, which now provide a consistent and robust taxonomy available to researchers and analysts. The second challenge, however, is far more complex. Much of the valuable, skill-related information is buried in unstructured text—such as online job advertisements—requiring advanced language models to extract meaningful insights. At the core of this process lies a crucial element: word embeddings, which unlock hidden patterns in language and transform raw text into actionable intelligence, as they enable machines to reason with text.

Generally speaking, Word embedding can be seen as a technique in natural language processing that represents words as dense vectors in a continuous vector space, capturing their semantic relationships and contextual meanings to improve machine understanding of language. As stated above, embeddings are crucial for identifying skills within a text. However, the process of creating embeddings is often arbitrary, leading to potentially biased outcomes. Different embedding models can produce significantly different results, making the selection of the right model essential; in fact, there is no established standard for selecting embeddings. This work tackles this issue by showing that different embedding choices can lead to dramatically different results and by proposing a novel framework for identifying the best embedding model for skill extraction. More specifically, we provide three main contributions to the literature. An overview of the research context and the main stages of the VEUCTOR framework is shown in Fig. 11.1.

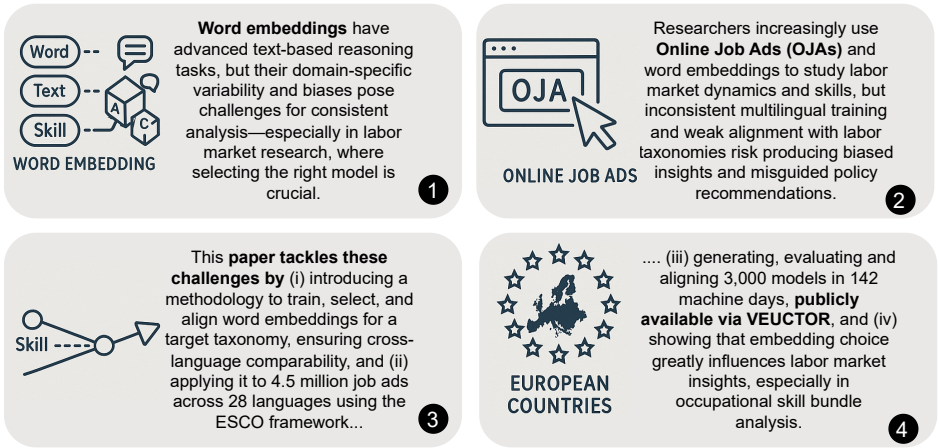


Figure 11.1: Graphical summary of the research context and the VEUCTOR framework.

Contribution. First, we define and implement a methodology to train and select vector space models that best fit a specific target taxonomy across multiple dimensions. We then align them to enable direct comparisons between models. The alignment process is particularly important in multi-language contexts such as the EU where language-specific issues may affect embeddings. Although the methodology is domain-independent, it is applied here within the field of labor and skill intelligence. It has been trained over 4.5 million job advertisements in 28 languages to build vector space models, us-

ing ESCO[5] as the target taxonomy and aligns country-specific embeddings to allow for cross-country comparisons.

Second, we synthesize over 3,000 embedding models for all 27 EU countries, plus the UK, identifying the best- and worst-performing embeddings with a total training time of 142 machine days. These models are aligned to the EU benchmark (i.e., the UK) and made available to the community as an off-the-shelf Python tool, namely VEUCTOR, allowing easy adoption and improving the reproducibility and comparability of different labor market analyses for the scientific community.

Third, we demonstrate the relevance of this work for real-world labor market analysis. We construct the occupational skill bundles derived from different embedding models, and we show that there is a *dramatic* difference in the skill bundles, highlighting how the choice of embedding model can significantly impact the conclusions drawn from skill analysis. This finding is even more significant following the number of highly relevant publications that use OJAs to conduct skill analysis.

More generally, although our work is applied to the EU labor market, our results can be generalized and extended to several other domains. The availability of a large amount of unstructured information contained in texts has led to a proliferation of tools, analyses, and research that process and analyze such data. For example, in Economics, there is a vast literature that has analyzed Central banks' communications and their effects on markets and expectations (see, for example, [240, 241]). What we show in this work is that the method of information extraction is not neutral, and the results can be highly dependent on it. This problem cannot be solved simply by more transparency, but must be accompanied by an optimization analysis such as the one we propose.

Roadmap. The methodology presented in this work follows a structured sequence of steps, which we summarize as follows:

1. Establishing a Benchmark – To evaluate the optimality of different word embeddings, a reference framework is needed for comparison. Since our embeddings are developed from a sample of Online Job Advertisements, we use the ESCO taxonomy as the natural benchmark.

2. Assessing Proximity to the benchmark – Once the benchmark is defined, a suitable methodology is required to measure the alignment of different embeddings with ESCO. We employ the Hierarchical Semantic Similarity (HSS) method for this analysis.

3. Addressing Multilingual Variability – Given the multilingual nature of our dataset, language differences can introduce distortions in embedding comparisons. To mitigate this, we implement an embedding alignment technique that ensures cross-language consistency.

4. Quantifying Differences – After evaluating the optimality of different embeddings, we measure the extent of variation between them. Specifically, we compare the best and worst embeddings by constructing a similarity metric for skill bundles. Our results indicate significant differences between data generated by the most and least optimal embeddings.

5. Providing data. Given the large differences in outcomes following different embeddings, we make available to the research community a data tool, VEUCTOR, which contains the codes and data for the best- and worst-generated embeddings, along with their optimality scores.

11.2 Building VEUCTORon 4 million OJAs and 28 EU Countries

Below, we present the workflow of embeddings generation and evaluation (Fig. 11.2) and alignment generation and evaluation (Fig. 11.3). Specifically, we introduce the framework for evaluating both the

[5]The ESCO taxonomy provides a structured representation of Skills, Competencies, Qualifications, and Occupations relevant to the European labor market. The European Commission has devised it to be a dictionary for the labor market in 27+1 countries and 32 languages. https://esco.ec.europa.eu/

embedding models trained for each country and the models obtained through the alignment process. The goal is to establish a systematic methodology for assessing and identifying the best-performing models using extrinsic evaluation measures. This evaluation framework allows us to determine the most and least effective models based on the conducted assessment. The process consists of several key steps, which we describe in detail in the following sections.

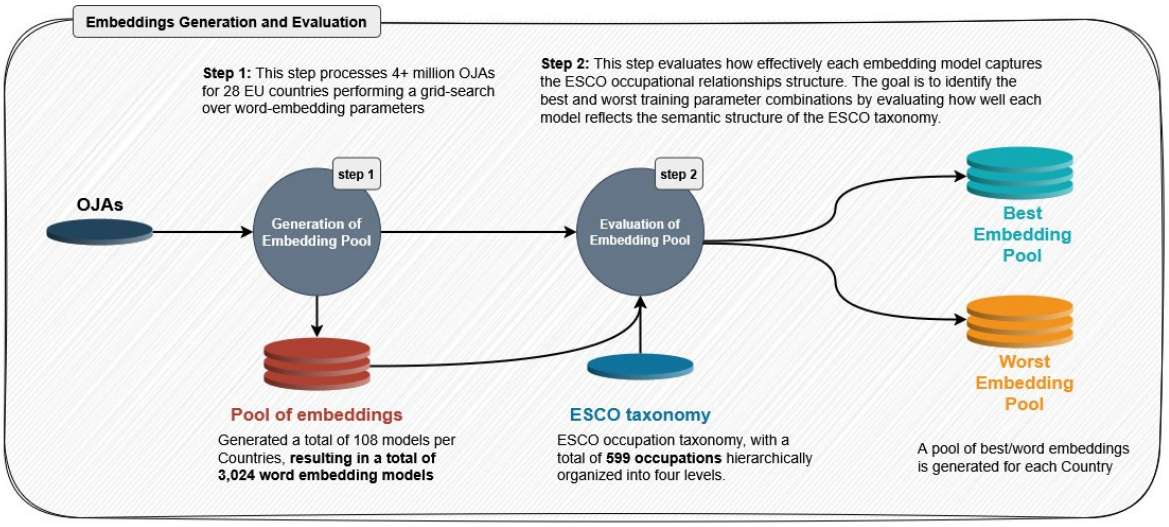


Figure 11.2: Workflow for the generation and evaluation of word embedding models.

Step 1 - Embeddings Pool Generation. This step consists of two main phases: i) preprocessing, where the raw text data from the OJA corpus is thoroughly cleaned, normalized, and prepared to ensure consistency across the dataset. The goal is to enhance the quality of the input data and make it suitable for further processing; ii) train different models for each country, in which semantic representations of words and phrases are derived from the preprocessed corpus. In this phase, FastText models are trained on the country-specific corpora using various combinations of training parameters. These embeddings serve as the semantic representations used in subsequent steps of the analysis.

Step 2 - Embeddings Pool Evaluation. In this step, we evaluate the quality of the embeddings generated in Step 1, focusing on each country individually. The evaluation is extrinsic and relies on the ESCO taxonomy as an external resource to assess how effectively the embedding models capture occupational relationships. The goal is to identify the optimal and suboptimal training parameter combinations by evaluating how well each model captures the semantic structure of the ESCO taxonomy.[6] A high-quality embedding model should not only preserve the real-world relationships between occupations but also maintain the hierarchical organization defined by ESCO. Specifically, their vector representations in the embedding space should exhibit high similarity if two occupations are closely related in ESCO—whether as siblings (i.e., sharing the same parent category) or in a direct parent-child relationship. Conversely, occupations that are distant within the hierarchy should show lower similarity. To assess the quality of the embeddings, we employed two key metrics: cosine similarity and Hierarchical Skill Similarity (HSS), the latter of which quantifies the semantic relationship between occupations in the ESCO taxonomy [242]. For every pair of occupations (ISCO 4-digit), we first calculate the cosine similarity between their embedding vectors. We then compute the HSS values for the same set of occupation pairs. Spearman's rank correlation coefficient (ρ) is applied to quantify

[6]**Our use of ESCO does not imply an endorsement of its superiority or optimality. Our approach is rather practical. ESCO is the official EU taxonomy and has become de facto the standard taxonomy used in Europe, making it the most logical and operationally viable benchmark for our purposes.**

the relationship between the cosine similarity scores and the HSS values. A higher correlation indicates a stronger alignment between the occupational concepts captured by the embedding models and the relationships defined within ESCO.

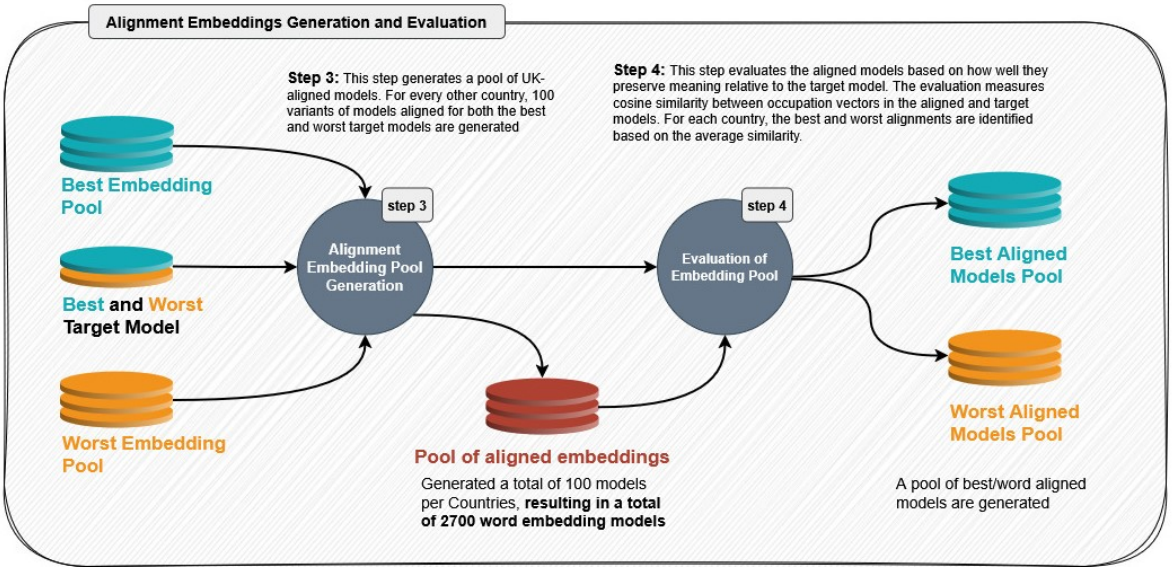


Figure 11.3: Workflow for the generation and evaluation of aligned models.

Step 3 - Alignment Embeddings Pool Generation. In this step, we align the embedding models using the SeNSe technique, as described in the previous section. The goal is to obtain comparable vector models across countries to enable inter-country analysis. The main challenge is that these models were trained on different corpora and with different sets of hyperparameters, making them inherently non-comparable. This discrepancy arises because the training process was not consistent across countries. To address this issue, we first select a common set of hyperparameters to standardize the models. Specifically, we choose the best-performing hyperparameter set based on the evaluation from the previous steps. This ensures that the 28 models (one per country) being aligned all share the same hyperparameter configuration, facilitating a more reliable comparison. Using an alignment technique, we transform the original vector spaces into a new set of aligned spaces, making them comparable. The SeNSe approach aligns source vector spaces to a predefined target space. In our case, we designate the UK model as the target space, meaning that all other country-specific vector spaces are aligned with the UK model. The alignment is performed pairwise, where each country's vector space acts as the source, and the UK model serves as the target. By aligning all 27 other countries to a common target, we ensure that any two countries can be directly compared within the same aligned space. The alignment algorithm requires tuning several parameters. In the previous step, we generated multiple alignment variations for different parameter combinations for each country, allowing us to assess the impact of different alignment configurations on model alignment.

Step 4 - Alignment Embeddings Pool Evaluation. In this step, we evaluate the aligned models for each country, which were generated in the previous stage. The evaluation criterion used is a cross-lingual semantic fitting score (*CLS score*): *Cross-Lingual* highlights that the comparison is performed between different languages, *Semantic* emphasizes that the evaluation focuses on preserving the meaning in the aligned vectors and *Fitting score* indicates that we measure how well the aligned model adapts to the target space. The evaluation assesses the aligned model based on its ability to generate similar vectors for the same concept when expressed in both the source and target models.

For this assessment, we consider occupations that appear in the vocabularies of both the source and target models. For each occupation, we compute the cosine similarity between the corresponding vectors in the aligned and target models. A higher cosine similarity indicates a stronger alignment between the concept in the source language and its counterpart in the target language. For each country, the best-aligned model is identified as the one that maximizes the average cosine similarity across all occupations. Similarly, the worst-aligned model is determined to have the lowest average cosine similarity. The input of the algorithm consists of the set of parameters used for alignment, the source models $\{\mathcal{M}_c\}c \in \mathcal{C}$ (where each model represents the best-performing model for a specific country c), and the list of occupations used for evaluation. For each country and each parameter combination, a new aligned model ($\mathcal{A}_{c,\theta}$) is generated. The cosine similarity is then computed between the vectors of the same occupation in the target model and the aligned model. The average cosine similarity across all occupations is calculated for each model, and the best/worst parameter combination for country c is selected as the one that maximizes/minimizes the similarity score ($score_{\mathcal{A}_{c,\theta}}$) among the different aligned models for c.

11.3 Experimental Results

The ESCO Taxonomy. ESCO (European Skills, Competences, Qualifications and Occupations) is the European classification of skills, competences, and occupations. It provides a multilingual dictionary of occupations and skill requirements organised along two main pillars. The first is the Occupation pillar, which is referenced tothe ISCO08 standard. The second is the Skills pillar, which lists and describes competencies/skills that are linked to occupations. Therefore, ESCO provides a list of occupations and related skills, organised as a network; nonetheless, it gives no information on the importance of skills in the considered occupation.

OJA Data. In the online job market, a job advertisement is a document containing two main text fields: a title and a full description. The title briefly summarizes the job position, while the full description field typically includes the position details and the relevant skills the employee must possess (see, e.g., [243]). The OJAs used here have been collected as part of the Web Intelligence Hub (WIH), which is a component of Eurostat's Trusted Smart Statistics (TSS) initiative, aiming to leverage web data for statistical purposes through the analysis of new data sources using advanced technologies such as artificial intelligence to integrate traditional sources used in official statistics. The WIH-OJA use case, developed by Cedefop and Eurostat and added to the WIH in 2021, is devoted to collecting and processing online job advertisements from sources in 32 countries (EU, EEA and the UK).

The OJA-WIH representative sample. Within the WIH initiative, Eurostat developed a representative sample of the entire OJA dataset, which is now composed of over 450 million unique OJAs collected since 2019. The sample aims to allow the community to work on a smaller dataset while preserving the distributional characteristics of online job ads. Specifically, the WIH-OJA-NLPv1 table seeks to leverage the richness of information extracted from online OJA portals. The goal of Natural Language Processing (NLP) data flows is to utilize the raw job titles and descriptions collected via web scraping techniques. The NLP samples serve two objectives, system development – a reference set of observations is necessary to test the impact of new techniques on the classification results – and research distribution – to enable research initiatives on OJA data. The sample is stratified by language, with the exception of very small ones and seven of the variables under which the OJAs are classified. The sampling is balanced and covers all possible values of the classification variables. The stratification variables include occupation (ISCO-08 III digit level), type of contract (permanent, temporary, apprenticeship or traineeship, and self-employed), salary, working time, education level, economic activity (NACE divisions), and experience. The v1 samples contain OJAs stratified by all

classified variables, and each stratum has a maximum of 50 observations, resulting in a total number of observations of 4,610,821. Considering the size of the sample and the level of detail provided, the database can be fruitfully employed to conduct research beyond the scope of quality. In this study, we use the NLP sample v1, release r20221217.

11.3.1 Data Pre-processing

In this step, raw text data undergoes several transformations to ensure consistency and remove noise that may affect subsequent analysis. We form the target texts by concatenating the job title with its description, as the title often contains relevant information such as the occupation or the required skills. First, unnecessary elements such as HTML tags, special characters, and URLs are removed, followed by the stripping of numerical values, certain symbols, and punctuation to retain only alphabetic content. The text is then normalized by converting all characters to lowercase, preventing inconsistencies due to case sensitivity. Additionally, sequences of characters representing meaningful multi-word expressions (n-grams) are identified[7] and replaced to preserve important phrases as single units. The preprocessing was further tailored to the corpus of each country, with linguistic operations adapted to the official languages of each (e.g., French, Dutch, and German for Belgium). Language-specific stopwords were excluded to refine the content, retaining only the most relevant terms. These steps collectively produce a clean and uniform dataset, improving its quality while enabling subsequent fastText models to better capture the corpus's concepts, identify linguistic relationships, and enhance advanced applications such as content analysis and creating rich semantic representations.

In addition to the preprocessing steps described above, Table 11.1 provides an overview of the OJAs datasets, detailing key statistics per country and year. The dataset covers 28 countries and includes information on the number of OJAs for 2020 and 2021, their total count, and the official languages used in each country. The table also highlights additional structural characteristics, including the number of distinct occupations, vocabulary size, and the average lengths of titles, descriptions, and tokens. These features highlight the dataset's heterogeneity and underscore the need for tailored preprocessing to extract meaningful patterns and enhance the performance of subsequent word embedding models in capturing semantic relationships and domain-specific concepts effectively.

11.3.2 Embeddings Pool Generation

In this section, we describe the process of generating, evaluating, and aligning embeddings for various occupations across different countries. The method produces occupational embeddings and evaluates their similarities using statistical metrics. The key steps of this process are outlined below.

In this stage, FastText models are trained on job advertisement data from 28 different countries to generate a diverse pool of embeddings for subsequent analysis. For each country, a separate corpus of job advertisements is processed, ensuring that linguistic and contextual differences are captured. The key component of this process involves hyperparameter optimization to assess how their variations impact the quality of the resulting word embeddings. A parameter grid search is employed to systematically explore different combinations of hyperparameters, including:

Embedding size (v_{size}): The dimensionality of the word vectors. Smaller vectors may lead to underfitting, where the embeddings are too simplistic to capture fine-grained semantic details, whereas larger vectors tend to capture more complex relationships but may require more data and computational resources.

Number of epochs (τ): The number of epochs was tested to strike a balance between training time and embedding quality. Lower values allow for faster training but may result in underfitting, where the model fails to capture the full complexity of the data. Higher values give the model

[7] `https://radimrehurek.com/gensim/models/phrases.html`

Table 11.1: Overview of OJAs statistics per country and year, including linguistic and structural features of the datasets.

Country	Country Code	Number of OJAs			Languages	Occupations	Vocabulary Size (# of words)	Text Lengths		
		2020	2021	Total				Avg. Title	Avg. Desc.	Avg. Tokens
Austria	AT	43,747	93,057	136,804	de	415	1,005,288	4.65	213.97	62.46
Belgium	BE	59,267	288,286	347,553	de, fr, nl	423	2,148,097	4.33	245.63	66.19
Bulgaria	BG	11,798	54,125	65,923	bg	525	670,812	9.14	281.95	74.86
Cyprus	CY	1,831	8,657	10,488	el	348	123,220	5.41	229.00	82.10
Czechia	CZ	7,223	56,598	63,821	cs	394	638,677	6.38	282.28	84.32
Germany	DE	96,779	391,060	487,839	de	492	2,955,779	5.63	246.48	62.92
Denmark	DK	8,302	38,625	46,927	da	369	728,429	6.81	470.79	146.84
Estonia	EE	1,919	13,742	15,661	et	337	157,323	3.65	142.46	53.85
Greece	EL	7,441	35,118	42,559	el	402	420,798	4.85	226.45	75.25
Spain	ES	43,003	192,681	235,684	es	419	1,170,647	5.22	233.57	58.12
Finland	FI	6,993	33,467	40,460	fi	395	403,520	3.386	192.604	74.603
France	FR	117,872	546,336	664,208	fr	510	2,865,121	5.651	294.121	70.191
Croatia	HR	6,093	30,874	36,967	hr	399	336,653	4.857	295.936	83.535
Hungary	HU	8,901	72,172	81,073	hu	403	796,307	4.509	227.544	72.518
Ireland	IE	26,698	94,760	121,458	en	409	720,802	4.173	304.217	85.605
Italy	IT	109,298	253,133	362,431	it	422	1,536,694	4.741	208.025	52.705
Lithuania	LT	5,279	25,229	30,508	lt	388	218,319	4.995	174.565	58.232
Luxembourg	LU	4,415	18,069	22,484	de, fr	326	193,320	5.854	274.225	76.536
Latvia	LV	4,950	21,201	26,151	lv	374	234,378	5.024	247.562	78.94
Malta	MT	1,176	5,925	7,101	mt	285	63,318	3.528	272.847	90.594
Netherlands	NL	60,613	316,420	377,033	nl	423	2,744,164	3.908	395.328	101.918
Poland	PL	48,400	122,337	170,737	pl	419	1,453,477	4.923	371.008	99.978
Portugal	PT	35,172	164,372	199,544	pt	449	1,679,525	5.274	307.137	80.549
Romania	RO	14,546	71,312	85,858	ro	411	612,775	4.854	231.257	67.846
Sweden	SE	28,410	136,420	164,830	sv	423	1,263,516	4.893	392.263	95.109
Slovenia	SI	2,495	11,221	13,716	sl	358	117,635	5.177	147.382	51.058
Slovakia	SK	8,349	55,080	63,429	sk	378	565,275	4.784	298.41	86.376
United Kingdom	UK	185,333	502,930	688,263	en	492	2,802,098	4.207	313.692	85.55
Total	28	921,738	3,497,491	4,419,229	23					

more opportunities to refine the embeddings and capture more nuanced semantic relationships, but they also increase the risk of overfitting.

Algorithm (A): This parameter defines the architecture of the neural network used during training. The Skip-Gram (SG) algorithm predicts context words from a given target word. It is particularly effective for smaller datasets, as it is better at capturing rare words and their semantic relationships. The Continuous Bag of Words (CBOW) algorithm predicts a target word based on its surrounding context. CBOW is generally more efficient for larger corpora, as it aggregates context words to make predictions, resulting in faster training times and improved performance on larger datasets.

Hierarchical softmax (*hs*): A parameter that determines whether hierarchical softmax is used, which can improve training efficiency when dealing with large vocabularies.

Learning rate (α): This parameter controls the speed at which the model updates its weights during training. A lower learning rate results in slower, more gradual updates, which can help the model converge more precisely; however, it may require more epochs to reach optimal performance. A higher learning rate enables faster updates, which can accelerate training but may lead to overshooting the optimal solution or instability in the learning process. The choice of learning rate has a significant impact on the balance between training time and the quality of the final embeddings.

For each country, 108 models were trained by varying the parameters across the following ranges: $v_{size} \in \{50, 100, 300\} \times \tau \in \{10, 50, 100\} \times \mathbf{A} \in \{SG, CBOW\} \times hs \in \{0, 1\} \times \alpha \in \{0.01, 0.05, 0.1\}$.

This was an exhaustive grid search without early stopping; each of the 108 parameter combinations per country was trained to completion to ensure a comprehensive evaluation.

11.3.3 Embeddings Pool Evaluation

Table 11.2: Best and worst FastText parameter combinations for each country, along with the corresponding Spearman correlation index (ρ) and its p-value (p_ρ). Training times of model pools for different countries are also given.

	Best Training Parameters							Worst Training Parameters							Machine Training Time (108 models per country)			
country	v_{size}	τ	A	*hs*	α	ρ	p_ρ	v_{size}	τ	A	*hs*	α	ρ	p_ρ	Avg.	Min	Max	Total Time
AT	300	10	SG	✓	0.1	0.088	2.481E-08	300	50	SG	✓	0.05	-0.069	1.034E-05	1h 40m	0h 10m	6h 35m	7d 12h
BE	300	100	SG	✓	0.1	0.247	1.652E-70	50	50	SG	✓	0.1	0.053	2.037E-04	1h 45m	0h 10m	6h 15m	7d 22h
BG	100	50	SG	✗	0.05	0.070	4.307E-04	300	100	CBOW	✓	0.1	-0.176	3.293E-20	0h 14m	0h 01m	0h 56m	1d 02h
CY	300	10	SG	✓	0.01	0.252	5.722E-81	50	100	CBOW	✗	0.1	0.001	9.149E-01	0h 13m	0h 01m	0h 56m	1d 00h
CZ	100	50	SG	✗	0.01	0.150	6.430E-14	100	50	CBOW	✓	0.1	0.005	7.877E-01	0h 18m	0h 01m	1h 11m	1d 09h
DE	100	100	SG	✓	0.1	0.143	8.582E-23	300	50	SG	✗	0.1	-0.060	4.624E-05	3h 17m	0h 17m	12h 02m	14d 19h
DK	300	50	CBOW	✓	0.01	0.256	9.118E-62	50	50	SG	✓	0.1	0.087	3.264E-08	1h 08m	0h 06m	4h 36m	5d 02h
EE	300	50	SG	✗	0.01	0.084	1.516E-10	100	100	CBOW	✓	0.05	-0.105	1.366E-15	0h 16m	0h 01m	1h 08m	1d 05h
EL	300	50	SG	✗	0.1	0.381	5.528E-167	300	100	CBOW	✗	0.1	0.150	9.620E-26	0h 23m	0h 02m	1h 37m	1d 18h
ES	300	10	SG	✗	0.05	0.335	6.348E-46	300	10	SG	✓	0.1	0.054	2.368E-02	1h 43m	0h 09m	7h 05m	7d 18h
FI	300	10	CBOW	✓	0.1	0.167	1.338E-28	50	100	CBOW	✓	0.1	-0.044	3.937E-03	0h 23m	0h 02m	1h 37m	1d 19h
FR	100	10	CBOW	✓	0.05	0.342	7.193E-42	50	10	CBOW	✗	0.01	0.170	4.893E-11	3h 21m	0h 17m	12h 32m	15d 02h
HR	300	10	CBOW	✗	0.01	0.285	4.971E-53	100	50	CBOW	✓	0.1	0.038	4.660E-02	0h 39m	0h 03m	2h 45m	2d 23h
HU	50	50	SG	✗	0.01	0.117	6.176E-11	300	100	CBOW	✗	0.1	-0.061	6.085E-04	0h 20m	0h 01m	1h 15m	1d 12h
IE	300	10	SG	✓	0.1	0.230	1.317E-56	50	10	SG	✓	0.01	-0.018	2.134E-01	1h 12m	0h 06m	5h 00m	5d 11h
IT	50	10	CBOW	✗	0.01	0.189	4.580E-27	100	10	CBOW	✓	0.1	-0.068	1.417E-04	1h 46m	0h 09m	6h 25m	7d 23h
LT	100	10	SG	✗	0.05	0.210	2.798E-28	50	100	CBOW	✓	0.05	-0.039	3.628E-02	0h 24m	0h 01m	1h 52m	1d 20h
LU	50	100	CBOW	✓	0.1	0.123	4.869E-12	100	10	SG	✓	0.1	-0.078	8.800E-06	0h 26m	0h 03m	1h 52m	1d 23h
LV	100	50	SG	✗	0.05	0.291	8.649E-43	50	50	CBOW	✓	0.1	0.011	6.094E-01	0h 28m	0h 02m	1h 58m	2d 03h
MT	300	50	SG	✓	0.1	0.338	5.270E-138	50	50	CBOW	✓	0.1	0.047	6.633E-04	0h 09m	0h 00m	0h 41m	0d 16h
NL	100	100	CBOW	✗	0.1	0.246	1.369E-41	100	100	CBOW	✓	0.01	0.121	3.405E-11	2h 13m	0h 13m	7h 28m	9d 23h
PL	50	10	CBOW	✗	0.01	0.291	1.181E-44	50	50	SG	✓	0.1	0.026	2.284E-01	1h 15m	0h 05m	4h 55m	5d 15h
PT	50	10	CBOW	✗	0.01	0.324	2.508E-64	300	100	SG	✗	0.1	0.053	7.035E-03	1h 05m	0h 07m	4h 44m	4d 21h
RO	300	100	SG	✗	0.01	0.220	2.204E-40	50	100	CBOW	✓	0.1	0.058	4.923E-04	0h 53m	0h 04m	3h 41m	4d 00h
SE	300	100	SG	✗	0.1	0.083	4.632E-08	100	10	SG	✓	0.01	-0.105	1.160E-12	0h 53m	0h 04m	3h 18m	4d 00h
SI	50	50	SG	✗	0.05	0.219	1.124E-22	50	50	CBOW	✓	0.05	0.013	5.677E-01	0h 21m	0h 02m	1h 34m	1d 13h
SK	300	10	SG	✗	0.05	0.170	3.319E-13	50	100	SG	✓	0.1	-0.017	4.775E-01	0h 29m	0h 01m	2h 29m	2d 05h
UK	50	50	CBOW	✗	0.01	0.269	1.017E-55	300	50	SG	✗	0.1	0.071	2.290E-05	3h 50m	0h 20m	14h 16m	17d 05h

This section describes the evaluation of the model pool trained for each country. The best and worst training parameter combinations were identified for each model and country using the Spearman rank correlation coefficient between the cosine similarity scores and HSS values for each occupation pair. Table 11.2 presents the evaluation results, showing the values corresponding to the training parameters used.

What emerges is that there is no single parameter combination that consistently yields the best results across different countries, as performance depends on both the chosen parameters and the training corpus. For example, for the selected algorithm, Skip-gram (SG) seems to perform the best in the majority of cases (18 countries), while Continuous Bag of Words (CBOW) appears to be the worst (16 countries). However, it is also observed that in some cases (12 countries), the algorithm is both the best and the worst, depending on the specific context.

It is also evident that the Spearman correlation values (ρ) are generally very low, with some of the worst cases even showing negative values. This indicates a significant task challenge, highlighting the difficulty of all models in accurately capturing the hierarchical relationships between occupations as defined in the ESCO taxonomy. Such low or negative correlation values may also reflect a gap between the real-world labor market and the structure of ESCO, suggesting that the embedding models struggle to fully reflect the semantic relationships inherent in the ESCO taxonomy.

Table 11.3: Comparison of pre-trained LLMs under different adaptation strategies. For each configuration (Zero-Shot, FT on ESCO, FT on OJA), the highest Spearman correlation (ρ) is underlined. The overall best result is highlighted in **bold**.

	LLM open-models	ρ	p_ρ
Zero-Shot	bge-large-en-v1.5	0.183	2.773E-28
	gte-large	0.193	2.207E-31
	bilingual-embedding-base	0.214	4.332E-38
	multilingual-e5-large-instruct	0.246	2.834E-50
Fine-Tuning su ESCO	bge-large-en-v1.5	0.178	1.088E-26
	gte-large	0.175	5.837E-26
	bilingual-embedding-base	0.198	5.255E-33
	multilingual-e5-large-instruct	0.217	2.042E-39
Fine-tuning su NLP Uk dataset	bge-large-en-v1.5	0.239	7.132E-95
	gte-large	0.211	1.861E-74
	bilingual-embedding-base	0.220	6.411E-81
	multilingual-e5-large-instruct	0.227	4.558E-86
	best UK-trained model (our)	**0.269**	1.017E-55

Comparison with Pretrained LLMs

A key question in the embedding-based analysis is whether a domain-specific model, trained directly on the OJAs corpus, can outperform large-scale pre-trained language models (LLMs) in capturing occupational similarities. While LLMs benefit from extensive training on diverse and heterogeneous corpora, their ability to accurately preserve structured relationships between occupations remains uncertain. To investigate this, we compare the best UK-trained embedding model against four pre-trained models, evaluating their performance using the same methodology adopted for pool evaluation.

To select appropriate pre-trained models for comparison, we relied on the benchmark constructed by [235], which provides a comprehensive evaluation of text embedding models. Their framework, the Massive Text Embedding Benchmark (MTEB), assesses models across various embedding tasks, offering insights into their relative strengths and weaknesses. MTEB evaluates models based on factors such as computational efficiency, embedding dimensionality, and performance on multiple NLP tasks. For our selection, we focused on the Semantic Textual Similarity (STS) task, as it closely aligns with our own evaluation methodology for assessing occupational similarity. The continuously updated leaderboard, ranking the best-performing models, is publicly available on Hugging Face platform[8]. According to the leaderboard, we select 4 open-source pre-trained models: *BAAI/bge-large-en-v1.5*[9] [233] with 335M params, *thenlper/gte-large*[10] [236] with 335M params, *Lajavaness/bilingual-embedding-base*[11] [244] with 278M params and *intfloat/multilingual-e5-large-instruct*[12] [245] with 7.11B params. This selection includes models based on key multilingual contextual architectures such as mBERT and XLM-R. To ensure a comprehensive and fair comparison, we evaluated each pre-trained model in three distinct settings:

1. **Zero-Shot:** Using the models out-of-the-box without any further training. This tests their inherent, general-purpose knowledge of occupational semantics.

2. **Fine-Tuned on ESCO:** We fine-tuned the models on the text descriptions of skills and occupations within the ESCO taxonomy. This tests their ability to adapt to the structure of our target

[8] https://huggingface.co/spaces/mteb/leaderboard
[9] https://huggingface.co/BAAI/bge-large-en-v1.5
[10] https://huggingface.co/thenlper/gte-large
[11] https://huggingface.co/Lajavaness/bilingual-embedding-base
[12] https://huggingface.co/intfloat/multilingual-e5-large-instruct

benchmark directly, providing them with explicit knowledge of the hierarchical relationships we aim to capture.

3. **Fine-Tuned on UK OJA Dataset:** We fine-tuned the models on the same corpus of UK job advertisements used to train our FastText models. This evaluates their performance when granted domain-specific adaptation on identical data, allowing for a direct and equitable comparison of modelling architectures by controlling for the training corpus.

Tab 11.3 shows the results of the comparison with the LLMs.The UK-trained model achieved the highest correlation score of 0.269, surpassing all four pre-trained models in terms of performance. Additionally, the p-values for all correlations are effectively zero, confirming the statistical significance of these results.

This demonstrates that even when LLMs are provided with the same domain-specific data (OJA fine-tuning) or explicit taxonomic knowledge (ESCO fine-tuning), our simpler, dedicated embedding approach better captures the hierarchical relationships defined in the ESCO taxonomy for this task. The advantage can be attributed to the fact that the FastText model was designed and trained directly to optimize geometric relationships in a vector space, which aligns well with the semantic similarity assessment performed by the HSS metric. Despite having orders of magnitude fewer parameters, its focused training objective proves more effective for this specific application than the more general-purpose representations of LLMs.

11.3.4 Alignment Embeddings Pool Generation

In this step, pools of embedding models aligned to the UK model are generated for each country. To align models using SeNSe, they must be trained with the same set of hyperparameters. As previously described, SeNSe aligns vector spaces by applying matrix transformations to shift the source space toward the target space, allowing the generation of new vectors for terms in the source space. For example, if the vector dimensions differ, this transformation cannot be applied, making prior standardization of hyperparameters essential. The best parameter combination across all countries was identified based on the highest average Spearman correlation. The optimal configuration consists of the following hyperparameters: $v_{size} = 300$, $\tau = 50$, $\mathbf{A}$ =SG, $hs = 0$, $\alpha = 0.01$. Therefore, only the 28 models trained with this configuration were considered for alignment. The UK model was chosen as the target for the alignment process, and all other country-specific models were aligned to it pairwise. This approach ensures that all models are mapped to a common space, enabling direct comparison across countries. SeNSe relies on several alignment parameters to optimize the transformation process. One of the most critical is the maximum allowed NDCG value for selecting the best anchors. The choice of this parameter significantly affects the quality of the alignment: lower thresholds result in fewer selected anchor points, meaning only the most confident semantic correspondences are used. While this can lead to a more stable transformation, it may also exclude useful but less obvious alignments. Higher thresholds increase the number of selected anchor points, leading to a denser alignment. This can help capture a broader range of semantic relationships but might introduce noise if less reliable correspondences are included. Given the potential impact of this parameter on alignment quality, it is crucial to systematically test different values. For this study, threshold values ranging from 0 to 0.99 were tested in increments of 0.01, generating a total of 100 different aligned models per country. This exhaustive approach ensures that we can identify the optimal alignment configuration for each country, balancing alignment accuracy and semantic consistency. Selecting the best-performing alignment model is essential, as a poor alignment could distort cross-lingual analyses, leading to misleading conclusions.

11.3.5 Alignment Embeddings Pool Evaluation

After generating multiple aligned models, as described in the previous section, their quality was assessed using the Cross-Lingual Semantic Fitting Score (CLS score). It aims to determine how well each aligned model preserves the semantic relationships of occupations relative to the target model. The evaluation process consists of the following steps: (i) the occupations present in both the source and target model vocabularies are considered, (ii) for each matched occupation, the cosine similarity between its vector representation in the aligned model and in the target model is computed and (iii) the average cosine similarity across all occupations serves as the final CLS score, indicating the overall alignment quality. Table 11.4 presents the evaluation results for each country's best and worst alignments. The table reports the NDCG threshold used to select anchor points during alignment, the CLS score measuring semantic consistency, and the confidence interval of the mean CLS score.

The results show a clear trend: the best alignments tend to have a lower NDCG threshold, meaning that a larger number of anchor points were used. This results in a more robust alignment, as more semantic relationships between the source and target spaces are preserved. Conversely, the worst alignments correspond to higher NDCG thresholds, which impose a stricter selection on anchor points. The few available anchors lead to a weaker alignment. These findings align with the work in [154], confirming that many anchor sets generally lead to better alignment performance.

A notable exception is observed for Ireland (IE), where the best alignment was obtained with a significantly higher NDCG threshold than in other countries. Consequently, its CLS score is also among the highest. This deviation can be explained by the shared official language (English) between Ireland and the UK, the target model. Since both vector spaces are already linguistically and semantically close, fewer anchor points are needed to achieve a high-quality alignment, reducing the need for a lower threshold.

These results further highlight the importance of carefully selecting the NDCG threshold for alignment. While a lower threshold generally improves alignment, language similarities between the source and target may allow for effective alignments even with fewer anchor points.

11.4 Assessing Skill Bundles across Europe using VECTOR

We analyse the best and worst-aligned embeddings after training, aligning, and evaluating all models. The objective is to assess how alignment quality affects occupational representations and the relationships between occupations and skills.

11.4.1 Occupation Representation

We first define how occupations are represented in the embedding space to analyze the differences between the best and worst alignment models in a specific occupational task. Rather than relying solely on the occupation term itself, we construct a more informative representation by leveraging the associated skill vectors. In essence, different embedding models are likely to generate different skill bundles, which are then likely to yield different results for the analysis.

For each occupation, we compute a centroid vector by averaging the embeddings of all the skills related to that occupation. Given an occupation o with a set of associated skills $S_o = \{s_1, s_2, ..., s_n\}$, where each skill s_i has an embedding $v(s_i)$, the centroid of the occupation is defined as:

$$v_o = \frac{1}{|S_o|} \sum_{s \in S_o} v(s) \tag{11.1}$$

In other words, v_o defines a vector representation of the skill bundle of occupation o. The vector representation is very useful as it allows one to compute measures of differences and similarities.

Table 11.4: Best and worst NDCG threshold for each country, along with the corresponding *CLS score*.

	Best Alignment Model			Worst Alignment Model		
country	NDCG threshold	CLS score	95% c.i	NDCG threshold	CLS score	95% c.i
AT	0.01	0.77	0.766 - 0.774	0.99	0.479	0.469 - 0.489
BE	0.0	0.809	0.805 - 0.814	0.99	0.428	0.418 - 0.438
BG	0.02	0.757	0.752 - 0.762	0.99	0.396	0.385 - 0.407
CY	0.03	0.7	0.695 - 0.705	0.99	0.444	0.433 - 0.455
CZ	0.02	0.749	0.743 - 0.756	0.99	0.441	0.427 - 0.455
DE	0.0	0.765	0.761 - 0.769	0.99	0.479	0.468 - 0.489
DK	0.0	0.761	0.755 - 0.768	0.99	0.41	0.395 - 0.424
EE	0.01	0.66	0.655 - 0.666	0.99	0.451	0.44 - 0.462
EL	0.02	0.774	0.769 - 0.778	0.99	0.442	0.432 - 0.451
ES	0.01	0.785	0.781 - 0.789	0.99	0.474	0.465 - 0.484
FI	0.01	0.727	0.722 - 0.732	0.99	0.446	0.436 - 0.456
FR	0.0	0.77	0.764 - 0.776	0.99	0.417	0.401 - 0.432
HR	0.02	0.727	0.72 - 0.733	0.99	0.454	0.442 - 0.467
HU	0.03	0.765	0.758 - 0.771	0.99	0.411	0.398 - 0.424
IE	0.26	0.835	0.83 - 0.84	0.99	0.392	0.377 - 0.406
IT	0.01	0.787	0.781 - 0.793	0.99	0.463	0.448 - 0.477
LT	0.01	0.725	0.718 - 0.732	0.99	0.485	0.47 - 0.5
LU	0.03	0.746	0.741 - 0.751	0.99	0.407	0.396 - 0.419
LV	0.01	0.696	0.688 - 0.703	0.99	0.478	0.465 - 0.491
MT	0.1	0.739	0.732 - 0.746	0.99	0.41	0.393 - 0.427
NL	0.0	0.798	0.792 - 0.804	0.99	0.429	0.416 - 0.443
PL	0.0	0.768	0.761 - 0.775	0.99	0.467	0.452 - 0.483
PT	0.02	0.812	0.806 - 0.817	0.99	0.443	0.428 - 0.458
RO	0.11	0.784	0.778 - 0.79	0.99	0.385	0.37 - 0.401
SE	0.0	0.778	0.772 - 0.784	0.99	0.417	0.404 - 0.429
SI	0.0	0.627	0.619 - 0.634	0.99	0.445	0.434 - 0.457
SK	0.03	0.734	0.729 - 0.738	0.99	0.445	0.435 - 0.455

11.4.2 Measuring skill similarities

To evaluate the potential bias introduced by model selection, we developed a simple indicator. Specifically, for each detailed occupation at the ISCO 4-digit level, we calculated a similarity measure for the associated skill bundle using the Jaccard distance between each identified skill and all other skills. Given the high level of occupational specificity, we expect the most effective embeddings to generate, within occupations, more internally consistent (i.e., similar) skill bundles compared to the least effective embeddings. Consequently, the magnitude of the difference in occupation skill similarity measures when comparing the best and worst embeddings provides an indicator of the bias introduced by model selection.

As emphasized in the previous paragraphs, the development of embeddings in a multilingual setting poses significant challenges related to the selection of the best embeddings and their alignment for comparability purposes. In principle, our results are affected by two elements: the optimization of embeddings and the optimization of the alignment procedure. In order to separate the two effects we start by performing an *intra*-country analysis, i.e. we compare *unaligned* models, focusing on differences between the best- and worst-performing models within each country. In this case, the analysis can be performed within countries, but the results cannot be compared between countries. Figure 11.4 displays the results. The left panel plots the cumulative distribution of the skill similarity measure by occupation of the best (blue line) and worst (red line) embeddings for a selection of countries (IT,

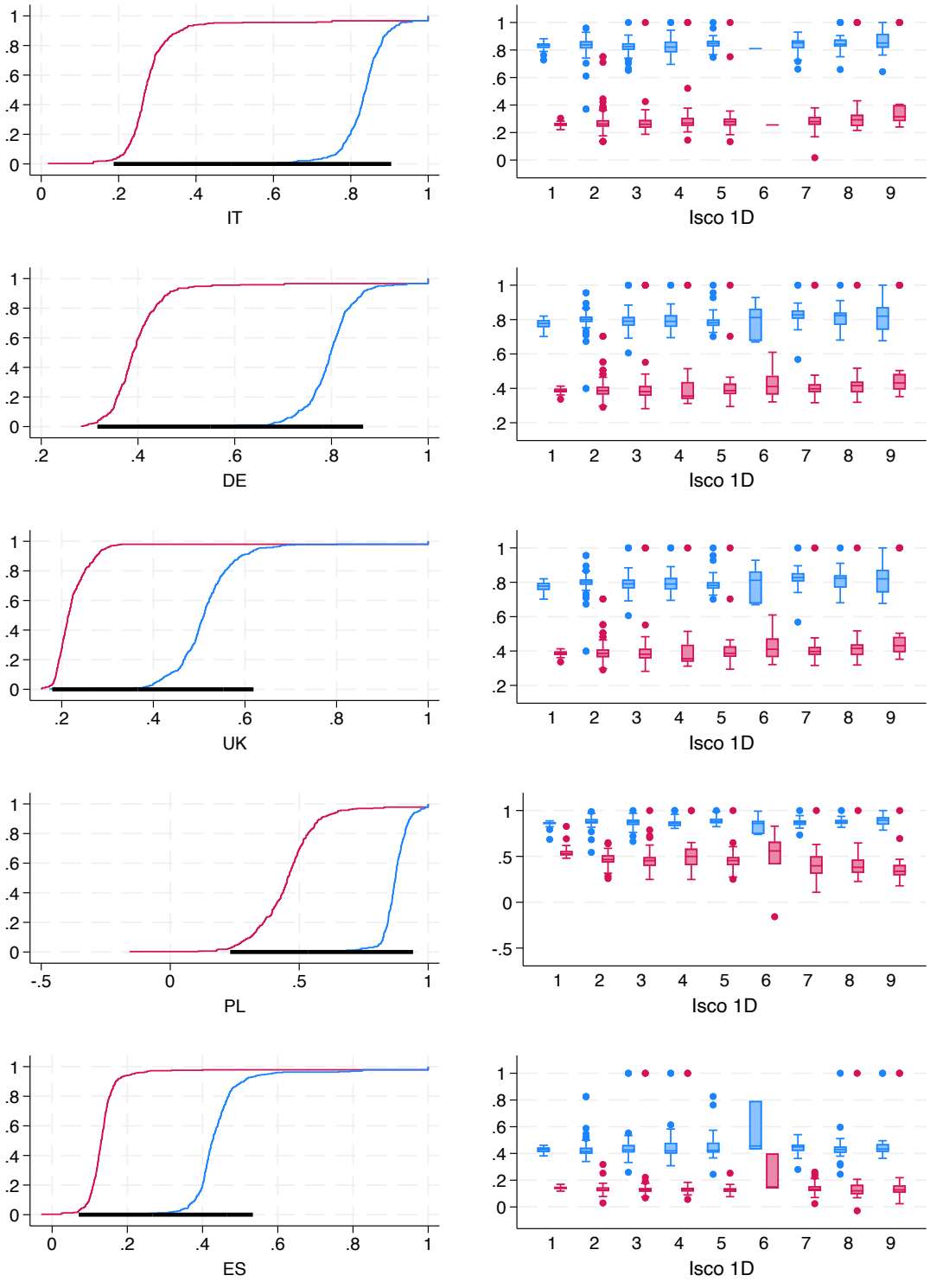
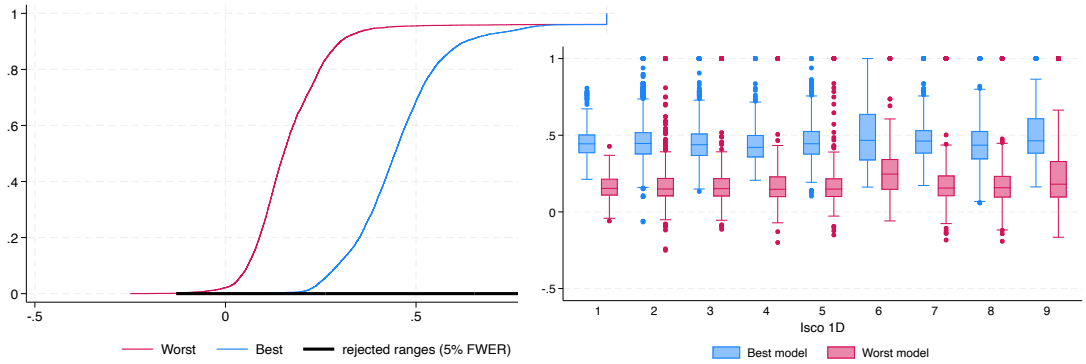


Figure 11.4: Distribution comparison: best and worst embeddings.

Best models: blue line/ boxplot. Worst model: red line/boxplot. Left panel: comparison of cumulative distributions of mean similarities of skill bundles generated with best and worst embeddings. The underlying black line denotes the region of rejection of the null hypothesis of equal distribution. Right panel: box plots of the distribution of mean similarity by occupation (1 digit ESCO)



(a) Distribution comparison best and worst models (b) Distribution of mean skill similarity over 1Digit ISCO

Figure 11.5: Left panel: comparison of cumulative distributions of mean similarities of skill bundles generated with best and worst embeddings. The underlying black line denotes the region of rejection of the null hypothesis of equal distribution. Right panel: box plots of the distribution of mean similarity by occupation (1 digit ESCO)

DE, UK, PL, ES). In both cases, the best embedding delivers a more similar set of skills by occupation. Beyond the eyeball metric offered by the two figures, we performed a formal test for the two distributions. The Goldman Kaplan [246] test rejects the equality of the two CDFs at 1%. The region of rejection of the familywise error rate is highlighted by the thick black line underlying the graph.[13] The right panel displays box-plots of the values of skill similarity by major occupation groups (1 Digit ESCO.). Similarly, there is a notable difference in the distribution of skill similarity by occupation between the two models, with the best embeddings yielding higher overall skill similarities across occupations.[14]

Therefore, our analysis has shown that the choice of embeddings can lead to significantly different skill bundles, ultimately influencing the outcomes of the analysis. Up to this point, the embeddings generated for different countries have not been aligned, restricting comparisons to within-country skill bundles. Next, we introduce the *alignment* procedure to enable cross-country comparisons of the results. Since both embeddings and alignment introduce degrees of freedom, we systematically explored all possible combinations by generating different skill bundles under the best and worst embedding conditions, as well as the best and worst alignment parameters.

This analysis yields two key findings. First, the alignment model produces highly robust results across different parameter specifications. Statistical tests fail to reject, at any significance level, the null hypothesis that the distribution of skill bundles is identical across different alignment configurations. This indicates that parameter selection for the alignment model does not introduce meaningful bias in the results.

Second, in contrast, the choice of embedding model proves to be of critical importance. Figure 11.5 presents a comparative analysis of the distribution of skill bundles across countries, contrasting the best and worst embeddings. The corresponding boxplot illustrates these differences within ISCO

[13] Instead of testing a single global null hypothesis (that the two CDFs are identical), the Goldman Kaplan methodology tests a continuum of individual null hypotheses of CDF equality at each point. This allows to obtain the ranges of x (if it exists) where $F(x) = G(x)$ is rejected at the specified error level. As the test is conducted for all points of the CDF, the methodology is exposed to the "multiple testing problem" that generates the "familywise error" where at least one true H_0 is rejected. Our approach achieves "strong control" of FWER, at a 10% level, meaning that there are zero false positives 90% of the time.

[14] The high variability of results for group 6 (Agricultural, Forestry and Fishery workers) is due low number of observations in OJAs for these occupations.

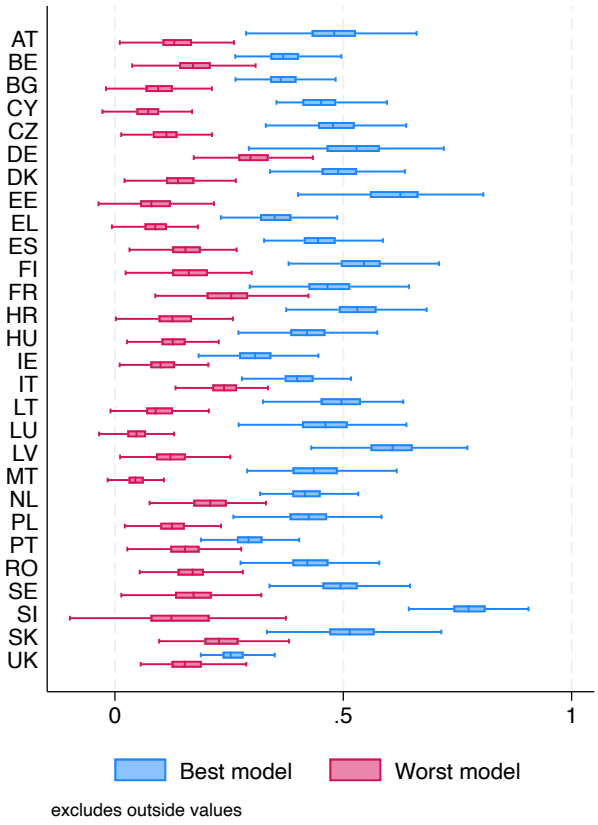


Figure 11.6: Distribution of mean skill similarity by country

1-digit occupational groups. Consistent with the intra-country results, cross-country comparisons further confirm that the best embeddings yield more similar skill bundles than the worst embeddings.

Figure 11.6 extends the cross-country analysis by comparing distributions across countries. The results demonstrate that the best embeddings consistently generate skill bundles that exhibit higher similarity across countries, reinforcing the robustness of this finding.

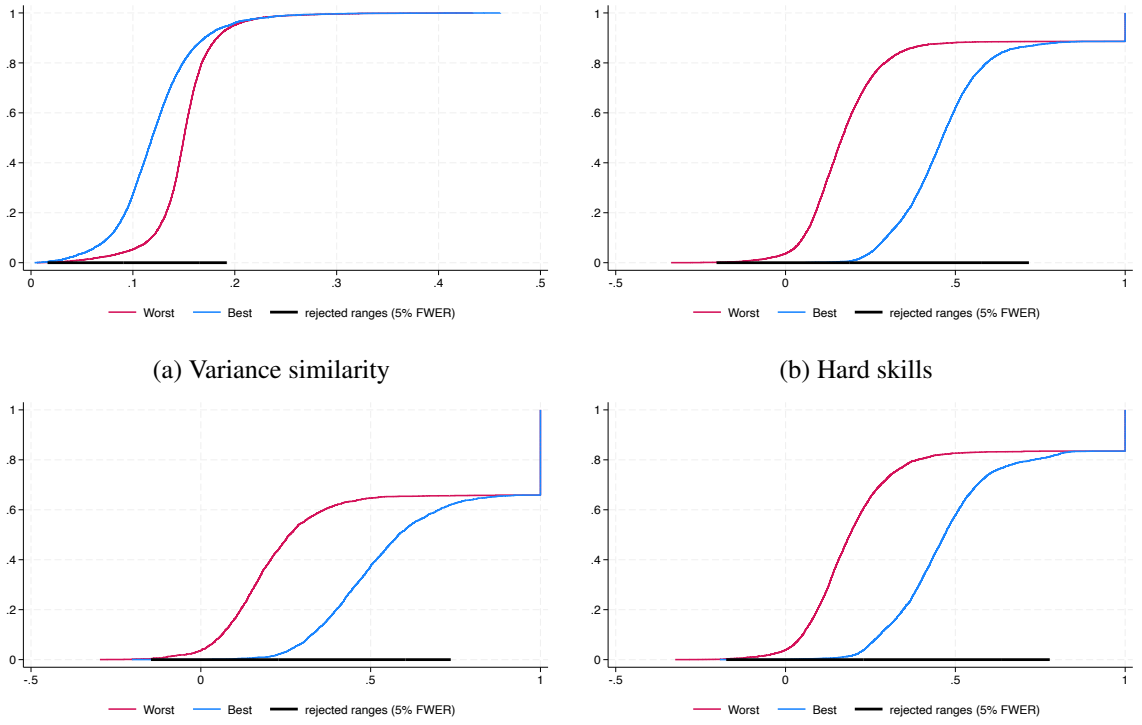
We conclude this section by providing some refinements to our analysis. We begin by considering not only the mean similarity across skill bundles but also its variance (Figure 11.7 panel a). As expected, the comparison of the two distributions shows that the variance of similarity measures is higher for the worst model than for the best ones. In other words, best models are characterized by more compact skill bundles, with both higher mean similarity and lower variance.

Finally, we have split the skill set into three categories — hard, soft, and digital skills — and performed the analysis separately for each group.[15] Figure 11.7, panels b, c, d show that the results obtained for the overall skill set are confirmed within each category: best models deliver more similar and statistically distinct skill bundles compared to the worst models.[16]

Overall, our results show that the selection of the embedding model generates a substantial and statistically significant variation in the distribution of skill bundles. On the contrary, the fine-tuning of alignment configurations does not have any statistically relevant effect.

[15]The categories are constructed using the ESCO classification.

[16]Note that in the digital skill set, there is a concentration at the end of the cumulative distribution. This is expected as the set of digital skills is limited, and several OJAs contain few general digital skills, such as Excel, where there is no difference between embedding models.



(a) Variance similarity

(b) Hard skills

(c) Digital skills

(d) Soft skills

Figure 11.7: Comparison of cumulative distributions of variance (panel a) and mean similarities (panels b, c, d) of skill bundles generated with best and worst embeddings. The underlying black line denotes the region of rejection of the null hypothesis of equal distribution.

11.4.3 Reproducibility

Data Availability. The codes, trained embedding models, and their aligned versions are made publicly available for research purposes. We provide pre-trained FastText embeddings for multiple countries, along with their aligned counterparts.[17]

Additionally, a GitLab repository has been created to facilitate reproducibility and further analysis. This repository contains the scripts and supplementary data used for evaluating both the original FastText models and their aligned versions.

```
import fasttext
model = fasttext.load_model('path/to/fasttext/model_best_uk')

from gensim.models.keyedvectors import Word2VecKeyedVectors
model = Word2VecKeyedVectors.load_word2vec_format('path/to/aligned/model_best_uk')
```

Listing 11.1: Example of loading the best model in a Python environment for UK.

In the Listing 11.1, an example of loading both a FastText model and an aligned model is provided. The official fasttext[18] library is used to load the model. For the aligned model, the widely used gensim[19] library is employed to load and utilize the model. Examples are provided using Python to align with the code used in our experiments; however, the choice of programming language is flexible. The FastText models are in `.bin` format, which can be loaded using various languages (e.g., Python,

[17] Due to the model sizes exceeding 150 GB, several Zenodo archives were created 10.5281/zenodo.15064201, 10.5281/zenodo.15064706, 10.5281/zenodo.15064719, 10.5281/zenodo.15064731, 10.5281/zenodo.15064738

[18] https://fasttext.cc/docs/en/python-module.html

[19] https://radimrehurek.com/gensim/

MATLAB, R, etc.), and the aligned models are provided in `.txt` format, making them even easier to handle.

Code Availability. All code used in this study was written in Python 3.12. The complete implementation is available in a GitLab repository (`https://gitlab.com/crisp1/veuctor.git`), which provides the scripts and additional data needed to reproduce our experiments.

Within the repository, the data folder contains supplementary resources, including the ESCO taxonomy for occupations and skills. Two Python scripts are included: `HSS_eval`, which implements the evaluation of the generated models using the HSS, and `alignment_eval`, which evaluates the aligned models. Additionally, a demo script is provided to illustrate the use of both models and the evaluation scripts. The repository also includes a README.md file with detailed instructions on use and setup.

Hardware Specifications and Processing Time. The computational experiments conducted in this study required significant processing power due to the large-scale training and alignment of embedding models across multiple countries. All computations were performed on an AWS cloud infrastructure to ensure efficiency and reproducibility.

The training of both the embedding models and their alignments was conducted using three machines,[20] each equipped with 32 CPUs and 64 GB of memory, along with one high-performance machine featuring 64 CPUs and 128 GB of memory.[21] The latter was specifically used for training models on larger corpora, such as those from the UK, France, and Germany. Training was parallelized both within each machine—distributing processes across available CPUs—and across multiple machines, with different countries assigned to different systems to optimize efficiency. All machines are equipped with an AMD EPYC 7R13 processor.

From Table 11.2, we observe that training times varied significantly across countries due to differences in corpus size and computational complexity. Countries with larger OJAs corpora, such as the UK, France, and Germany, required substantially more time for model convergence. Additionally, the total training time for all models combined exceeded 140 days, highlighting the scale of the computational effort involved.

The alignment process was considerably faster than training, as it involved matrix transformations rather than learning word representations from scratch. The generation and evaluation of the aligned model pool were completed in approximately 2 days and 10 hours.

11.5 Concluding Remarks

Every fisherman knows that the type of fish caught crucially depends on the bait that is used. In our context, not considering the possible bias generated by the choice of the embedding model is equivalent to inferring the composition of the fish population in a lake based on what is caught, failing to recognize that the type of bait heavily affects the catch.

In this work, we have demonstrated that the choice of word embeddings significantly influences the type of information extracted from both structured and unstructured text. To address this, we developed a methodology to evaluate and compare different word embeddings, identifying the most effective one for a given task. Additionally, we introduced a framework for aligning word embeddings across multiple languages, enhancing their applicability in multilingual contexts.

Applying our methodology to a comprehensive dataset of Online Job Advertisements in Europe, we showed that different word embeddings yield distinct informational outputs, leading to substantially different skill bundle categorizations. As a key contribution to the research community, we

[20] AWS 3x c6a.8xlarge
[21] AWS c6a.16xlarge

provide VEUCTOR, a tool that not only implements our methodology but also offers access to pre-computed word embeddings. This resource enables researchers to conduct labor market analyses with an optimized information extraction approach.

Although our study focuses on the European labor market, our findings have broader implications, extending to any domain that relies on embeddings to extract insights from textual data. This is particularly relevant in the social sciences, where the increasing availability of large-scale unstructured text has driven a proliferation of analytical tools and research methodologies. We, therefore, advocate for the adoption of our approach in various fields, facilitating more accurate and context-aware analyses.

Part VI

Vector Space Model for Finance domain

Chapter 12

Evaluating the Effectiveness of Fine-Tuning in Financial NLP: The Case of Social Trading Action Detection

The impact of online information dissemination on financial markets is well-established in the literature, with investor sentiment, emerging trends, and sector-specific discussions demonstrating significant influence on stock market dynamics. However, most existing methods for this purpose rely on simple sentiment analysis models, which fail to capture the concrete trading intentions expressed in these discussions. While Large Language Models (LLMs) offer a promising alternative to simplistic sentiment analysis, the actual benefits of fine-tuning across different model families remain unclear in noisy, domain-specific contexts like online forums.

This chapter introduces and formalizes a novel task: *Social Trading Action Detection*, a novel task classifying online posts into actionable categories (buy, sell, or other), and present FINREDDIT-2K[1], a manually annotated dataset of 2,123 Reddit posts, designed to serve as a benchmark for this task.

Our investigation establishes vector space modeling as the foundational framework for this task, evaluating a comprehensive spectrum of natural language processing approaches: (i) nine traditional neural network classifiers; (ii) three zero-shot sequence classifiers; (iii) twenty-three Large Language Models (LLMs) evaluated in zero-shot settings; and (iv) twenty-two LLMs evaluated in fine-tuned settings.

The results demonstrate the clear superiority of approaches leveraging rich semantic representations, with fine-tuned LLMs achieving remarkable performance. Specifically, Mistral-7B attains the highest F1-score (86.0%), followed by Neural-chat-7B (84.7%) and Phi-4-14B (84.6%). This comprehensive experiment provides an in-depth evaluation of the benefits and limitations of fine-tuning, highlighting not only the types of errors it can mitigate but also those it may introduce.

This work is supported by the Italian Ministry of University and Research (MUR) within the PRIN2022—ISALDI: Interpretable Stock Analysis Leveraging Deep multi-modal models (CUP: E53D 23008150006).

12.1 Introduction

Large Language Models (LLMs) have demonstrated strong effectiveness across many NLP tasks in the financial domain [104, 16, 247], particularly when adapted through parameter-efficient fine-tuning methods such as LoRA [248]. The majority of existing studies, however, focus on identifying the best-performing model, which may rapidly change given the velocity of progress in this field, rather than

[1] Under final review on Information Processing & Management Journal

developing a deeper understanding of how fine-tuning influences model behavior in specific domains.

In business contexts, organizations may not always be able to adopt the model that has been reported in the literature as the top performer for a specific task. This limitation may arise from several factors, such as licensing constraints, costs, resource availability, or the coexistence of different model families. In such cases, a company may attempt to specialize a model it already uses, with the expectation that fine-tuning should improve its performance compared to the base model. However, this assumption has not been consistently confirmed, as several studies have shown that fine-tuning does not always lead to significant gains [249, 250, 251].

In this work, we perform a detailed analysis of the effect of fine-tuning on a variety of LLMs for a novel financial NLP task involving the analysis of noisy social media posts. In addition to identifying the best-performing architectures, we examine how fine-tuning interacts with different model characteristics and analyse the types of errors it alleviates or amplifies. The results of our study provide deeper insights into the strengths and limitations of fine-tuning when applied to a financial NLP task.

12.1.1 Social Trading Action Detection

It is now widely recognized that the dissemination and exchange of online information can exert a substantial influence on economic dynamics, particularly with respect to stock prices and trading volumes [104]. The rapid growth of digital platforms has provided numerous arenas where investors, traders, and other stakeholders actively discuss stocks, financial markets, and investment strategies [110]. Platforms such as Yahoo Finance, StockTwits, InvestorHub, and Reddit have become central venues for these interactions [16]. These dynamics enable large groups of retail investors to coordinate their behavior through online platforms, either emerging spontaneously or by rallying around influential social media figures whose posts serve as focal points for collective decision-making. Such coordinated behavior can significantly influence financial markets. The most prominent example is the GameStop case [252], which showed how coordinated retail investors could challenge traditional market mechanisms. To fully harness the informational potential of online platforms, it is essential to accurately identify and interpret investment-related suggestions within user-generated content. This task is commonly addressed as Financial Sentiment Analysis (FSA), where each post about a particular stock is assigned a positive or negative label. Consequently, automated sentiment analysis tools have become widely adopted in this domain due to their accessibility, ease of deployment, and the growing body of supporting research [253, 254, 13].

However, despite their popularity, FSA methods have a strong limitation: they extract the general sentiment of a text rather than capturing the specific action performed or recommended by the user with respect to a particular stock. Although a positive sentiment may sometimes align with a recommendation to purchase a stock, this correspondence is not consistent. For instance, a post may express positive sentiment toward a company's values without suggesting a purchase, or it may criticise a stock while still recommending buying it for strategic reasons. A notable example occurred during the GameStop case, where negatively worded posts could indicate either a "buy" action as a form of protest against institutional investors or a "sell" recommendation to avoid anticipated losses. Consequently, identical sentiment labels could correspond to opposing investment strategies depending on the context.

In order to capture actionable signals emerging from financial online communities without reducing them to the oversimplified categories of sentiment analysis, it is necessary to directly interpret the concrete suggestions expressed in user posts. To this end, this chapter introduces and investigates a novel task, *Social Trading Action Detection* (STAD), which involves classifying online posts according to the user's expressed action or recommendation regarding the intention to buy or sell a specific stock. We formalise STAD as a multi-class classification problem in which each post is assigned to one of three categories: *buy*, *sell*, or *other*. This task presents significant challenges due to the

complexity and ambiguity of financial online discussions and cannot be effectively addressed through standard sentiment analysis techniques, as demonstrated in the evaluation presented in this chapter.

Large Language Models (LLMs) have demonstrated remarkable performance on tasks requiring advanced natural language understanding, including the analysis of financial documents [112, 255, 256], which makes them promising candidates for addressing STAD. Nonetheless, these models also exhibit persistent weaknesses that may hinder their effectiveness in interpreting financial discussions. Common limitations include hallucinations, difficulties in managing specialised jargon, and reduced robustness when dealing with noisy, ambiguous, or sarcastic text [257, 258]. These shortcomings are particularly relevant to STAD, since online financial discourse often relies on idiomatic expressions, irony, and community-specific terminology. Such content poses challenges even for human readers who are not embedded in these communities.

For these reasons, we argue that STAD represents a rigorous and practically relevant testbed for evaluating both the strengths and the limitations of next-generation LLMs. The task requires fine-grained linguistic understanding, resilience to noisy and domain-specific language, and the ability to distinguish actionable recommendations from general commentary. It therefore provides an opportunity to assess LLM performance in a challenging real-world setting, while also paving the way toward enhancing their capacity to support decision-making in complex social and financial environments.

12.1.2 Research questions

In this work, we present a comprehensive study on the application of LLMs to the STAD task. Our objective is not limited to determining whether LLMs can outperform alternative methods or identifying the most effective LLM-based architecture. Rather, we provide a critical assessment of the performance exhibited by the LLMs under consideration, with special attention to the extent to which fine-tuning contributes to performance gains. Furthermore, we investigate their ability to handle linguistic phenomena that frequently occur in online posts, such as implicit language and sarcasm.

The analysis was driven by the following research questions:

- *RQ1:* How effective are modern LLMs in performing STAD compared with traditional models, and what is the best-performing LLM Model?
- *RQ2:* Does fine-tuning LLMs on high-quality data significantly enhance their performance on this task?
- *RQ3:* To what extent does the effectiveness of fine-tuning depend on the characteristics of the underlying model?
- *RQ4:* What types of errors occur most frequently, and how does fine-tuning influence their distribution, severity, and nature?

To address these research questions, we present FINREDDIT-2K, a new dataset including 2,123 Reddit posts that have been manually classified by domain experts into three predefined categories. The data were gathered from multiple finance-related subreddits and span a broad range of stocks, thereby ensuring both thematic diversity and close alignment with real discussions in the financial domain.

To answer *RQ1*, we compared LLMs performance with that of traditional approaches, including nine neural network classifiers such as the Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and bidirectional LSTM (Bi-LSTM), along with three zero-shot sequence classifiers.

RQ2 and *RQ3* motivated a comprehensive evaluation of multiple models. To address these questions, we conducted a systematic assessment of LLMs on FINREDDIT-2K, testing twenty-three models in zero-shot settings and twenty-two models in fine-tuned settings. These experiments enable a

critical analysis of model performance and provide insights into the extent to which fine-tuning enhances results across different architectures.

Finally, to answer *RQ4*, we conducted an in-depth error analysis of the misclassifications produced by the models and the underlying factors contributing to them. We also grouped the errors into distinct categories that reveal the current limitations of foundational models in this domain. These limitations include difficulties in distinguishing between a generic option and an explicitly recommended action, challenges in interpreting sarcasm and slang, failures in differentiating the reporting of news from an actual suggestion, and confusion when multiple action descriptions occur within the same post.

12.1.3 Main contributions

This work advances research in financial NLP for information extraction from social media through two main contributions. First, it formalizes STAD and introduces a comprehensive benchmark designed to support systematic model evaluation. Second, it provides a set of insights, with particular emphasis on the effects of fine-tuning large language models in this challenging domain.

The experimental evidence shows that contemporary LLMs can markedly surpass traditional models in STAD, provided that training is carried out on high-quality datasets. By contrast, zero-shot LLMs frequently perform worse than a strong MLP baseline, reinforcing the conclusion that fine-tuning is indispensable for unlocking their full potential. The extent of the improvements, however, varies substantially across architectures. Instruction-tuned medium-sized models (e.g., Mistral-7B) yield the most effective trade-off between accuracy and efficiency, whereas certain architectures fail to benefit and may even experience degradation after fine-tuning. The error analysis indicates that fine-tuning systematically mitigates critical errors such as action inversions and biased false positives, shifting the overall error profile toward less harmful misclassifications. Yet, ambiguity, irony, and implicit language remain persistent difficulties, highlighting the need for more sophisticated contextual reasoning. In summary, our findings show that carefully designed fine-tuning strategies, together with the selection of robust model families, are fundamental for achieving state-of-the-art results in this complex task. In summary, the main contributions of this work are as follows:

- We adopt the task of *Social Trading Action Detection* as a testbed for evaluating a wide range of LLMs and provide an in-depth investigation of the impact of fine-tuning and the types of errors produced.
- We introduce and release FINREDDIT-2K, a novel dataset for financial post classification that contains 2,123 manually annotated Reddit posts categorized into three trading actions. The dataset is publicly available as a repository[2].
- We benchmark LLMs against 12 traditional baselines, showing that they achieve superior performance on this task when fine-tuned.
- We conduct a detailed study on the effect of fine-tuning, demonstrating both its contribution to performance improvement and its influence on model behavior and error distributions.
- We provide an in-depth analysis that identifies, quantifies, and discusses the typical limitations of LLMs in interpreting online posts.

12.2 Methodology

In this section, we describe the methodology employed to investigate LLMs on STAD. Our approach, illustrated in Figure 12.1, involves the construction of a benchmark dataset followed by the evaluation

[2] https://github.com/Simone-Damico/FinReddit-2K-STAD

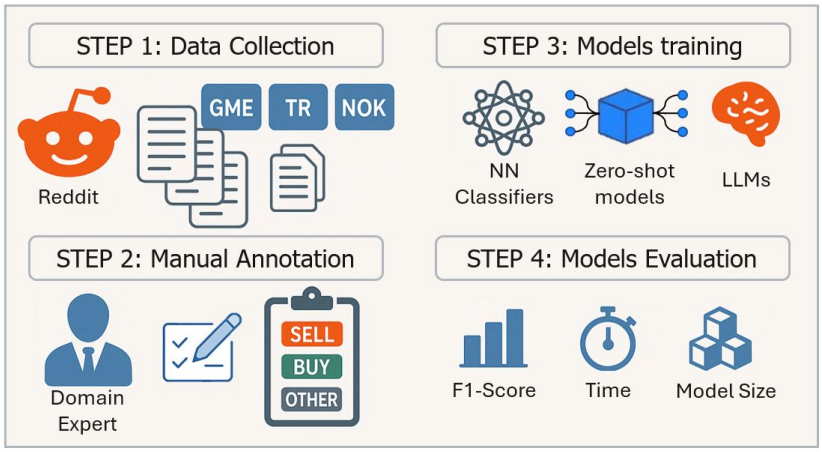


Figure 12.1: Workflow of the proposed methodology.

of multiple families of classification methods on this benchmark. We first provide a formal definition of the STAD tasks, and then describe each stage of the experimental methodology.

12.2.1 Task Definition

We formulate STAD as a multi-class classification problem: for a given natural language input, the objective is to categorize it into one of three classes: *buy*, *sell*, or *other*. More, formally, we define STAD as a function f that maps a textual input x (a social media post) to one of three predefined class labels y in the set $\mathcal{Y}$:

$$f : x \to y \quad \text{where} \quad y \in \mathcal{Y} = \{\texttt{buy}, \texttt{sell}, \texttt{other}\} \tag{12.1}$$

The labels are defined as follows:

- `buy`: The post explicitly suggests, implicitly encourages, or affirms the author's intention to *purchase* the stock.
- `sell`: The post explicitly suggests, implicitly encourages, or affirms the author's intention to *divest* or *short* the stock.
- `other`: The post discusses the stock but does not advocate for a specific trading action. This includes neutral commentary, news reporting, factual questions, or off-topic discussion.

This formalization allows us to tackle the problem with standard classification methodologies. Nevertheless, the task introduces distinct challenges stemming from the informal, context-dependent, and frequently nuanced nature of language used on social media. User-generated content often incorporates domain-specific jargon, sarcasm, or idiomatic expressions, all of which necessitate a profound semantic comprehension and specialized domain knowledge for correct interpretation.

12.2.2 The FinReddit-2K dataset

To support the training and evaluation of classifiers for the action categorization task (step 2 as shown in Figure 12.1, we introduce FinReddit-2K, a manually annotated benchmark dataset for the Social Trading Action Detection (STAD) task. The dataset comprises 2,123 Reddit posts labeled into three actionable categories: *buy*, *sell*, or *other*. Below, we detail its construction and key characteristics.

Data Source Identification. Reddit is a popular social media platform organized into thematic communities known as *subreddits*, which facilitate focused user discussions and content sharing. For this study, we targeted eight finance-oriented subreddits: *wallstreetbets*, *finance*, *economics*, *investing*,

Table 12.1: Distribution of posts collected from Reddit by stock in the FINREDDIT-2K dataset

Stock	**Ticker**	**# Posts**	**Coverage (%)**
GameStop	GME	979	46.114%
Nokia	NOK	166	7.819%
Tootsie Roll	TR	132	6.218%
Aurora Cannabis	ACB	125	5.888%
AMC Entertainment Holdings	AMC	124	5.841%
Hims & Hers Health	HIMS	123	5.794%
General Motors	GM	123	5.794%
First Majestic Silver	AG	120	5.652%
BlackBerry	BB	118	5.558%
Rocket Companies	RKT	113	5.323%
Total	10	**2,123**	**100%**

pennystocks, *StockMarket*, *Stocks*, and *Dividends*. These communities serve as central hubs for financial discourse, where users actively exchange views on stock price movements, macroeconomic trends, and investment strategies. Consequently, they represent a rich and relevant source of data for investigating retail investor sentiment and collective decision-making processes in digital environments.

To ensure a comprehensive analysis of the relationship between online discourse and investor sentiment, we selected ten stocks (listed in Table 12.1) designed to capture both the distinctive dynamics of the meme stock phenomenon and a diverse cross-section of market sectors. GameStop (GME) and AMC Entertainment (AMC) were chosen as emblematic examples of meme stocks, noted for their high volatility and substantial retail investor communities, making them ideal for studying social media-driven market effects. The remaining stocks were selected to provide variety in industry coverage and investment profiles: BlackBerry (BB) and Nokia (NOK) represent legacy technology firms in transition; First Majestic Silver (AG) offers exposure to the commodities sector; Hims & Hers Health (HIMS) and Rocket Companies (RKT) reflect emerging industries like telehealth and fintech; General Motors (GM) exemplifies traditional automotive manufacturing adapting to innovation; Aurora Cannabis (ACB) covers the speculative cannabis market; and Tootsie Roll (TR) provides a contrast as a stable consumer goods brand.

Data Collection and Selection. To collect posts related to these ten stocks, we developed a Python-based module that leverages the PRAW (Python Reddit API Wrapper)[3] library to retrieve Reddit posts that met specific criteria. Posts were included if either the title or the body text contained the full name or ticker symbol of any of the selected stocks. Each Reddit post was associated with a single stock. We selected a total of 3,000 posts spanning the period from *March 20, 2008* to *July 9, 2024*. The dataset was balanced so that 50% of the posts focused on GameStop, while the remaining 50% covered the other nine stocks.

Annotation Process. Two domain experts manually annotated the posts. The annotators were instructed to perform two tasks: (i) to classify each post into one of three categories, and (ii) tdentify and exclude posts that were off-topic, generated by bots, or too ambiguous for reliable classification due to low relevance or high ambiguity [259]. Each annotator labeled the posts independently. Upon completion of the independent labeling, the agreement between the experts was 87%. In cases where the assigned labels did not match, the annotators discussed the discrepancies to reach a consensus and

determine the appropriate label from among those initially proposed.

Dataset Statistics and Characteristics. The final dataset contains **2,123** posts. Key statistics are as follows:

- Label Distribution: *Buy* (54.0%), *Sell* (9.5%), *Other* (36.5%). This reflects a realistic class imbalance in financial social media.
- Text Length: Posts have an average length of **160 words**, ranging from 6 to 5,032 words.
- Temporal Span: Covers over 16 years of financial discussions (2008–2024).
- Stock Coverage: The distribution across the ten stocks is detailed in Table 12.1.

12.2.3 Adopted Models

In this section, we discuss in detail the four categories of approaches evaluated on the FINREDDIT-2K dataset.

Neural network classifiers. We evaluated the performance of three traditional architectures on STAD: MLP, LSTM, and Bi-LSTM. Before training, we applied a preprocessing step to the texts to remove irrelevant tokens. Specifically, low-relevance elements such as web addresses and Reddit-specific markers like 'r/' and 'u/'—which indicate references to subreddits and users—were removed. Additionally, the label '[deleted]' was eliminated, as it typically appears in place of removed users or content, ensuring that only meaningful text was retained for classification.

To identify an optimal text representation strategy, we leveraged the Massive Text Embedding Benchmark (MTEB) [235], a comprehensive framework for evaluating text embedding models across a diverse suite of NLP tasks. MTEB facilitates comparison of over 300 models based on criteria including performance, runtime efficiency, and embedding dimensionality. Based on this evaluation, we selected three open models from the Hugging Face platform to serve as encoders for the Reddit posts, prioritizing a balance between model performance and computational efficiency. The selected models are: all-MiniLM-L6-v2[4] (22.7M params), all-mpnet-base-v2[5] (109M params), and gte-large-en-v1.5[6] [260] (434M params). These encoders were used to transform the preprocessed Reddit posts into fixed-dimensional vector representations, which subsequently served as input features for the neural network classifiers.

Pretrained zero-shot sequence classifiers. Breakthroughs in transfer learning enabled the use of large pre-trained language models for downstream tasks with minimal or no additional training [25, 261]. In this context, zero-shot sequence classification has emerged as a powerful methodology, allowing models to perform categorisation tasks without requiring domain-specific labeled data.

In our study, we employed three widely used transformer-based models for zero-shot learning. The first, mDeBERTa-v3-base-mnli-xnli[7] (279M params), incorporates disentangled attention mechanisms to enhance contextual understanding. The second, bart-large-mnli[8] (407M params), is a sequence-to-sequence model pre-trained on large-scale corpora and fine-tuned on the Multi-Genre Natural Language Inference (MNLI) dataset. The last model is xlm-roberta-large-xnli[9] (561M parameters), which extends the xlm-roberta-large model by fine-tuning it on a combination of natural language inference

[3] https://praw.readthedocs.io/en/stable/
[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[5] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[6] https://huggingface.co/Alibaba-NLP/gte-large-en-v1.5
[7] https://huggingface.co/MoritzLaurer/mDeBERTa-v3-base-mnli-xnli
[8] https://huggingface.co/facebook/bart-large-mnli
[9] https://huggingface.co/joeddav/xlm-roberta-large-xnli

Table 12.2: Overview of selected LLMs with release year, family, size category, and training methodology. All the models are avalible on the Huggging Face platform.

Model	Size	Family	Release year	Training Methodology
GPT-2-XL	Small (1.6B)	GPT-2	Pre-2024	Base
Falcon3-7B	Small (7B)	Falcon	2024	Base
Gemma-2-7B	Small (7B)	Gemma	2024	Instruction-Tuned
Mistral-7B	Small (7B)	Mistral	2024	Instruction-Tuned
Neural-chat-7b-v3-1	Small (7B)	Mistral	Pre-2024	Instruction-Tuned
Nous-Hermes-2-7B	Small (7B)	Mistral	2024	Instruction-Tuned
OLMo-7B [262]	Small (7B)	OLMo	2024	Instruction-Tuned
Qwen-Distill-DeepSeek-R1 (7B)	Small (7B)	Qwen	2025	Distilled
Qwen2.5-7B	Small (7B)	Qwen	2024	Instruction-Tuned
Starling-LM-7B [263]	Small (7B)	Mistral	2024	Instruction-Tuned
Zephyr-7B [264]	Small (7B)	Mistral	Pre-2024	Instruction-Tuned
LLaMA-3.1-8B	Small (8B)	LLaMA	2024	Instruction-Tuned
LLaMA-Distill-DeepSeek-R1 (8B)	Small (8B)	LLaMA	2024	Distilled
Falcon3-10B	Medium (10B)	Falcon	2024	Instruction-Tuned
SOLAR-10.7B	Medium (10.7B)	Mistral	Pre-2024	Instruction-Tuned
Gemma-3-12B	Medium (12B)	Gemma	2025	Instruction-Tuned
FinGPT-v3.3-13B [265]	Medium (13B)	LLaMA	Pre-2024	Instruction-Tuned
LLaMA-2-13B	Medium (13B)	LLaMA	Pre-2024	Instruction-Tuned
Qwen1.5-14B	Medium (14B)	Qwen	2024	Base
Phi-4-14B	Medium (14B)	Phi	2024	Base
Gemma-3-27B	Large (27B)	Gemma	2025	Instruction-Tuned
QwQ-32B	Large (32B)	Qwen	2025	Instruction-Tuned
LLaMA-3.3-70B	Large (70B)	LLaMA	2024	Instruction-Tuned

datasets. All models are designed for Natural Language Inference, enabling them to classify posts by evaluating their semantic similarity to predefined category descriptions.

LLMs in zero-shot setting. We assessed the efficacy of Large Language Models (LLMs) for zero-shot learning (ZSL) text classification utilizing a prompt-based inference paradigm. Each Reddit post was presented to the model alongside a fixed instructional prompt that directed it to categorize the content into one of the three predefined labels. This methodology capitalizes on the robust generalization abilities inherent in LLMs, which, by virtue of their pre-training on extensive and diverse corpora, can interpret and execute natural language instructions without requiring any task-specific fine-tuning.

Our evaluation encompassed a broad spectrum of 23 contemporary open-weight LLMs, selected to represent a wide range of architectures, sizes, release eras, and training methodologies as shown in Table 12.2. This diverse selection enables a comprehensive analysis of how different model characteristics influence ZSL performance on the STAD task.

1. **Model Size:** Small (<10B), Medium (10B-20B), Large (>20B), reflecting parameter count and expected representational capacity.

2. **Architectural Family:** Grouped by model lineage, including LLaMA, Mistral-based models, Gemma, Falcon, Qwen, and Unique architectures covering hybrids and specialized models.

3. **Release Era:** Pre-2024, 2024, 2025, capturing the evolution of training techniques and pre-training corpora.

4. **Training Methodology:** Base (pre-trained only), Instruction-Tuned (SFT on instruction-response data), and Distilled (student models trained from larger teachers).

Fine-tuned LLMs. A subset of the LLMs previously evaluated in the zero-shot learning (ZSL) setting was selected for fine-tuning on the FINREDDIT-2K dataset. This selection was designed to ensure variation across both model architectures and scales. To guarantee a consistent task definition, we retained the identical instructional prompt utilized in the zero-shot experiments. Fine-tuning was performed via Low-Rank Adaptation (LoRA) [248], which injects trainable low-rank matrices into frozen pre-trained weights. This parameter-efficient approach accelerates training, reduces computational requirements, and preserves the model's original knowledge, making it ideal for domain adaptation.

Furthermore, we extended our fine-tuning analysis to include BERT-based architectures, which have established a strong precedent for performance in text classification tasks [266]. We evaluated BERT-base (110M parameters) and BERT-large (340M parameters). Following previous studies in the financial domain [110], we categorize BERT as an LLM, acknowledging this classification is debatable.

12.2.4 Experimental Analysis

In this section, we describe the comparative evaluation of the previously presented methods using the FINREDDIT-2K dataset. All experiments were conducted on a machine equipped with an NVIDIA A100-SXM4 with 80GB of GPU memory.

Experimental protocol. We conducted a 5-fold cross-validation on FINREDDIT-2K using the models described in the previous section. These models include nine neural network classifiers (MLP, LSTM, and Bi-LSTM, each employing the three encoder models introduced in Section 12.2.3), three zero-shot sequence classifiers, 23 LLMs in ZSL setting, and 21 fine-tuned LLMs.

The experimental protocol comprises the following steps:

1. To prepare the textual data for downstream analysis and enhance the models' ability to distinguish between the three classes, we applied a pre-processing pipeline aimed at reducing noise and standardizing input. Specifically, we removed HTML tags, URLs, Reddit-specific markers (e.g., /r/, u/, [deleted]), and all type of whitespace characters. This cleaning process ensures that the resulting text is free from noise and better suited for subsequent modeling steps.

2. For neural network classifiers and BERT-based models, we conducted a grid search over their model-specific hyperparameters. These hyperparameters include batch size, hidden layer dimensions, dropout rate, maximum sequence length, and the number of training epochs. Other models were trained only once, using specific values for parameters such as training epochs, optimizer, or temperature for generation.

3. For each model and each hyperparameter combination, we performed a stratified k-fold cross-validation with $k = 5$. This procedure ensures reliable performance estimation while maintaining the original class distribution within each fold. During training, 10% of the training data was reserved as a validation set.

Given the imbalance among the target classes, we employ weighted versions of standard evaluation metrics to ensure a more accurate assessment of the model's performance. Specifically, we calculate the weighted *precision*, *recall*, and *F1-score*, which take into account the number of samples in each class. In addition, we report the overall *F1-score* and *accuracy*.

While global scores like accuracy or F1-score provide a general sense of quality, they fail to reveal *what kinds of mistakes* the model makes and *how costly* they are in practical terms. For instance,

misclassifying a clear *sell* signal as *buy* may have significantly worse consequences than confusing it with a neutral *other*. Hence, we perform a structured error analysis from different points of view. First, we analyse consistent patterns of misclassification. Beyond the standard error categories, namely Type I (false positives) and Type II (false negatives), we also distinguish three additional domain-specific types:

- *Action Inversion Errors:* when a *buy* is predicted as *sell*, or vice versa.
- *False Action Errors:* when a neutral post (*other*) is misclassified as an action (*buy* or *sell*).
- *Non-action Errors:* when a post that should trigger an action (*buy* or *sell*) is predicted as neutral (*other*).

Second, we conduct a qualitative analysis of the best model predictions, considering both correctly and incorrectly classified posts. For each case, we also asked the model to generate an explanation of its own prediction, which we used to assess the consistency between the predicted label, the underlying text, and the model's reasoning. This procedure allows us to highlight linguistic patterns and contextual ambiguities that drive both accurate classifications and systematic errors, providing additional insight beyond aggregate metrics.

Evaluation settings. In the following, we report the evaluation settings used for the four families of approaches.

- **Neural network classifiers:** For each model, we perform a grid search over the following parameter combinations: *encoder* $\in$ {*all-MiniLM-L6-v2, all-mpnet-base-v2, gte-large-en-v1.5*}$\times$ *model* $\in$ {*MLP, LSTM, Bi-LSTM*}$\times$ *hidden dimension layer* $\in \{64, 128\} \times$ *batch size* $\in \{16, 32, 64\}$ $\times$ *dropout* $\in \{0, 0.05, 0.1\} \times$ *max sequence lenght* $\in \{64, 128, 256\}$ $\times$ *training epochs* $\in \{5, 10\}$ for a total of 972 generated models.
- **Pretrained zero-shot sequence classifiers:** We use the three models described in Sec. 12.2.3 directly in their original form, without additional training or fine-tuning.
- **LLMs in zero shot setting:** All models were deployed using a 8-bit quantization technique specifically designed to reduce memory consumption and speed up inference. For the generation step, to ensure reproducibility, we set the temperature to 0.001.
- **Fine-tuned LLMs:** A grid search was conducted to fine-tune BERT using the following parameters: *model* $\in$ {*BERT-base, BERT-large*}$\times$ *batch size* $\in \{8, 16, 32\} \times$ *max sequence lenght* $\in \{128, 256, 512\} \times$ *training epochs* $\in \{2, 4\}$, for a total of 36 generated models. For the other LLMs, as for zero-shot, we used 8-bit quantization and a temperature of 0.001. Additionally, for the training phase, the following parameters were set: LoRA attention dimension (r parameter) = 64, batch size= 8, learning rate = 2e-4, number of training epochs = 2 and AdamW optimizer.

12.3 Results

In the following, we present and discuss the results of the experiments. For clarity, the analysis is organized according to the four research questions defined in the introduction.

RQ1 - Performance of LLMs against Traditional Models

Concerning RQ1, which investigates how the performance of LLMs compares with traditional solutions, Table 12.3 presents the results of the 57 evaluated models. The pre-trained zero-shot sequence

Table 12.3: Model evaluation results. Each metric is reported along with its 95% confidence interval, computed over the 5-fold cross-validation. Models are ranked by F1 for each family. The best model for each family is underlined; the overall best model is shown in bold.

Family	Model	Accuracy		Precision		Recall		F1-score	
		Value	C.I.95%	Value	C.I.95%	Value	C.I.95%	Value	C.I.95%
Neural network classifiers	gte-large-en-v1,5 + MLP	0.729	0.724 - 0.734	0.673	0.647 - 0.699	0.729	0.724 - 0.734	0.696	0.685 - 0.707
	all-MiniLM-L6-v2 + MLP	0.674	0.667 - 0.681	0.628	0.623 - 0.633	0.674	0.667 - 0.681	0.625	0.616 - 0.634
	all-mpnet-base-v2 + MLP	0.687	0.676 - 0.698	0.631	0.621 - 0.641	0.687	0.676 - 0.698	0.644	0.632 - 0.656
	all-mpnet-base-v2 + LSTM	0.540	0.539 - 0.541	0.292	0.291 - 0.293	0.540	0.539 - 0.541	0.379	0.378 - 0.380
	all-MiniLM-L6-v2 + LSTM	0.535	0.518 - 0.552	0.447	0.389 - 0.505	0.535	0.518 - 0.552	0.419	0.397 - 0.441
	gte-large-en-v1,5 + LSTM	0.540	0.539 - 0.541	0.292	0.291 - 0.293	0.540	0.539 - 0.541	0.379	0.378 - 0.380
	gte-large-en-v1,5 + Bi-LSTM	0.605	0.561 - 0.649	0.544	0.418 - 0.670	0.605	0.561 - 0.649	0.530	0.432 - 0.628
	all-MiniLM-L6-v2 + Bi-LSTM	0.540	0.539 - 0.541	0.292	0.291 - 0.293	0.540	0.539 - 0.541	0.379	0.378 - 0.380
	all-mpnet-base-v2 + Bi-LSTM	0.540	0.539 - 0.541	0.292	0.291 - 0.293	0.540	0.539 - 0.541	0.379	0.378 - 0.380
Pretrained zero-shot sequence classifiers	bart-large-mnli	0.535	0.522 - 0.548	0.617	0.598 - 0.636	0.535	0.522 - 0.548	0.549	0.536 - 0.562
	xlm-roberta-large-xnli	0.508	0.496 - 0.520	0.636	0.619 - 0.653	0.508	0.496 - 0.520	0.516	0.502 - 0.530
	mDeBERTa-v3-base-mnli-xnli	0.513	0.494 - 0.532	0.643	0.609 - 0.677	0.513	0.494 - 0.532	0.496	0.474 - 0.518
LLMs in zero-shot setting	gemma-3-27B	0.694	0.681 - 0.707	0.704	0.691 - 0.717	0.694	0.681 - 0.707	0.682	0.668 - 0.696
	gemma-3-12B	0.613	0.587 - 0.639	0.651	0.615 - 0.687	0.613	0.587 - 0.639	0.600	0.573 - 0.627
	Falcon3-10B	0.589	0.574 - 0.604	0.560	0.544 - 0.576	0.589	0.574 - 0.604	0.548	0.530- 0.566
	SOLAR-10.7B	0.586	0.572 - 0.600	0.586	0.565 - 0.607	0.586	0.572 - 0.600	0.543	0.523 - 0.563
	Llama-2-13b	0.592	0.575 - 0.609	0.592	0.567 - 0.617	0.592	0.575 - 0.609	0.538	0.512 - 0.564
	Falcon3-7B	0.582	0.563 - 0.601	0.566	0.542 - 0.590	0.582	0.563 - 0.601	0.538	0.521 - 0.555
	Starling-LM-7B	0.584	0.571 - 0.597	0.582	0.562 - 0.602	0.584	0.571 - 0.597	0.535	0.520 - 0.550
	Llama-3.1-8B	0.569	0.558 - 0.580	0.570	0.548 - 0.592	0.569	0.558 - 0.580	0.526	0.515 - 0.537
	OLMo-7B	0.515	0.488 - 0.542	0.525	0.504 - 0.546	0.515	0.488 - 0.542	0.513	0.490 - 0.536
	Llama-3.3-70B	0.497	0.469 - 0.525	0.558	0.526 - 0.590	0.497	0.469 - 0.525	0.510	0.482 - 0.538
	Mistral-7B	0.558	0.542 - 0.574	0.542	0.509 - 0.575	0.558	0.542 - 0.574	0.505	0.483 - 0.527
	Nous-Hermes-2-7B	0.558	0.543 - 0.573	0.539	0.510 - 0.568	0.558	0.543 - 0.573	0.504	0.487 - 0.521
	DeepSeek-R1-Distill-Llama-8B	0.517	0.497 - 0.537	0.497	0.469 - 0.525	0.517	0.497 - 0.537	0.482	0.459 - 0.505
	Qwen2.5-7B	0.514	0.493 - 0.535	0.547	0.521 - 0.573	0.514	0.493 - 0.535	0.480	0.460 - 0.500
	gpt2-xl-1.6B	0.525	0.515 - 0.535	0.520	0.492 - 0.548	0.525	0.515 - 0.535	0.472	0.458 - 0.486
	phi-4-14B	0.492	0.480 - 0.504	0.542	0.508 - 0.576	0.492	0.480 - 0.504	0.431	0.423 - 0.439
	QwQ-32B	0.428	0.404 - 0.452	0.533	0.513 - 0.553	0.428	0.404 - 0.452	0.424	0.404 - 0.444
	gemma-7b	0.358	0.330 - 0.386	0.538	0.514 - 0.562	0.358	0.330 - 0.386	0.384	0.358 - 0.410
	neural-chat-7b	0.349	0.335 - 0.363	0.642	0.623 - 0.661	0.349	0.335 - 0.363	0.383	0.366 - 0.400
	DeepSeek-R1-Distill-Qwen-7B	0.407	0.393 - 0.421	0.487	0.468 - 0.506	0.407	0.393 - 0.421	0.360	0.341 - 0.379
	Qwen1.5-14B	0.272	0.253 - 0.291	0.558	0.48 - 0.636	0.272	0.253 - 0.291	0.225	0.206 - 0.244
	zephyr-7B	0.227	0.218 - 0.236	0.371	0.329 - 0.413	0.227	0.218 - 0.236	0.160	0.155 - 0.165
	FinGPT v3.3-13B	0.096	0.094 - 0.098	0.225	0.000 - 0.484	0.096	0.094 - 0.098	0.018	0.015 - 0.021
Fine-tuned LLMs	Mistral-7B	0.860	0.844 - 0.876	0.862	0.846 - 0.878	0.860	0.844 - 0.876	**0.860**	0.844 - 0.876
	neural-chat-7B	0.848	0.833 - 0.863	0.849	0.832 - 0.866	0.848	0.833 - 0.863	0.847	0.831 - 0.863
	phi-4 - 14B	0.847	0.835 - 0.859	0.847	0.835 - 0.859	0.847	0.835 - 0.859	0.846	0.834 - 0.858
	zephyr-7B	0.840	0.811 - 0.869	0.844	0.821 - 0.867	0.840	0.811 - 0.869	0.840	0.812 - 0.868
	Llama-3.1-8B	0.821	0.805 - 0.837	0.832	0.815 - 0.849	0.821	0.805 - 0.837	0.822	0.807 - 0.837
	bert-base-uncased	0.746	0.739 - 0.753	0.741	0.731 - 0.751	0.746	0.739 - 0.753	0.737	0.728 - 0.746
	bert-large-uncased	0.709	0.692 - 0.726	0.649	0.614 - 0.684	0.709	0.692 - 0.726	0.669	0.640 - 0.698
	SOLAR-10.7B	0.581	0.554 - 0.608	0.604	0.571 - 0.637	0.581	0.554 - 0.608	0.574	0.545 - 0.603
	FinGPT v3.3-13B	0.566	0.536 - 0.596	0.593	0.554 - 0.632	0.566	0.536 - 0.596	0.548	0.513 - 0.583
	Nous-Hermes-2-7B	0.572	0.554 - 0.590	0.588	0.557 - 0.619	0.572	0.554 - 0.590	0.548	0.525 - 0.571
	Llama-2-13B	0.569	0.537 - 0.601	0.588	0.547 - 0.629	0.569	0.537 - 0.601	0.546	0.504 - 0.588
	Starling-LM-7B	0.570	0.547 - 0.593	0.581	0.546 - 0.616	0.570	0.547 - 0.593	0.531	0.507 - 0.555
	OLMo-7B	0.568	0.286 - 0.850	0.629	0.320 - 0.938	0.568	0.286 - 0.850	0.529	0.259 - 0.799
	DeepSeek-R1-Distill-Llama-8B	0.567	0.551 - 0.583	0.555	0.532 - 0.578	0.567	0.551 - 0.583	0.513	0.495 - 0.531
	Qwen1.5-14B	0.504	0.100 - 0.908	0.504	0.100 - 0.908	0.504	0.100 - 0.908	0.504	0.101 - 0.907
	Falcon3-10B	0.575	0.560 - 0.590	0.570	0.545 - 0.595	0.575	0.560 - 0.590	0.492	0.470 - 0.514
	Falcon3-7B	0.564	0.552 - 0.576	0.552	0.508 - 0.596	0.564	0.552 - 0.576	0.475	0.462 - 0.488
	Qwen2.5-7B	0.491	0.479 - 0.503	0.574	0.549 - 0.599	0.491	0.479 - 0.503	0.442	0.429 - 0.455
	DeepSeek-R1-Distill-Qwen-7B	0.530	0.518 - 0.542	0.522	0.477 - 0.567	0.530	0.518 - 0.542	0.437	0.422 - 0.452
	gpt2-xl-1.6B	0.539	0.531 - 0.547	0.488	0.460 - 0.516	0.539	0.531 - 0.547	0.425	0.415 - 0.435
	QwQ-32B	0.487	0.467 - 0.507	0.549	0.525 - 0.573	0.487	0.467 - 0.507	0.422	0.403 - 0.441
	gemma-7b	0.366	0.314 - 0.418	0.569	0.529 - 0.609	0.366	0.314 - 0.418	0.383	0.34 - 0.426

classifiers, together with the LSTM and Bi-LSTM baselines, achieve relatively low F1-scores, only slightly above 50%. In contrast, the MLP model combined with the gte-large-en-v1.5 encoder attains a substantially higher F1-score (69.6%). This result outperforms all large language models in the zero-shot setting and suggests that, without fine-tuning, large language models are not necessarily the most effective solution. Table 12.4 presents the optimal parameter configurations for the baseline models identified through a grid search. The best-performing baseline model is gte-large-en-v1.5 combined with an MLP, using the following parameter configuration: 10 epochs, batch size of 16, dropout set to 0, input size of 64, and hidden size of 64.

Table 12.4: Best parameter combinations for each neural network classifier and BERT models.

Model	Epochs	Batch size	Dropout	Input size	Hidden size
all-MiniLM-L6-v2+MLP	10	16	0	256	128
all-mpnet-base-v2+MLP	10	16	0	128	128
gte-large-en-v1,5+MLP	10	16	0	64	64
all-MiniLM-L6-v2+LSTM	5	32	0.1	64	128
all-mpnet-base-v2+LSTM	5	64	0	256	128
gte-large-en-v1,5+LSTM	10	64	0	256	64
all-MiniLM-L6-v2+Bi-LSTM	5	32	0.1	128	128
all-mpnet-base-v2+Bi-LSTM	10	32	0.1	256	128
gte-large-en-v1,5+Bi-LSTM	10	32	0.1	256	128
BERT-base	4	16	0.1	256	768
BERT-large	4	32	0.1	256	1024

The results of the LLMs in the zero-shot scenario show considerable variation. Gemma-3-27B delivers the best performance, reaching an F1-score of 68.2%. Conversely, several systems, such as QwQ-32B, score below 50%. A particularly striking case is FinGPT 13B, a model explicitly designed for financial sentiment classification, which performs very poorly. This outcome reinforces the observation that STAD cannot be adequately tackled by relying on standard sentiment analysis approaches.

In the fine-tuning scenario, the best-performing model is Mistral-7B, which achieves an F1-score of 86.0%. The top five fine-tuned models (Mistral-7B, neural-chat-7B, phi-4, zephyr-7B, and Llama-3.1-8B) all exceed the 80% threshold, confirming the effectiveness of fine-tuning for this task. Within the BERT family, the base version proves the most effective, reaching 73.7%, while the larger variants do not yield substantial improvements.

These results indicate that both the complexity of the model and the application of task-specific fine-tuning play a crucial role in attaining strong performance. Although lightweight architectures like MLPs can achieve competitive outcomes when combined with high-quality sentence embeddings, LLMs show clear benefits once fine-tuned to the target task. In addition, parameter-efficient strategies such as LoRA allow effective fine-tuning without incurring excessive computational overhead, reinforcing recent directions toward scalable and widely accessible fine-tuning approaches [248]. The performance gains observed through fine-tuning are consistent with earlier studies (e.g., [110]), underscoring the importance of domain adaptation for fully exploiting the potential of pre-trained LLMs [267, 268].

RQ2 - Benefits of Fine-Tuning

Table 12.5 presents a comparison of different LLMs by reporting their average inference and fine-tuning times, together with the variation in F1-score when transitioning from zero-shot to fine-tuned configurations. As previously discussed, fine-tuning proved to be highly effective, increasing the average F1-score by approximately 15.1%. Seven models achieve an F1 improvement of more than 25%. Among these, five (Zephyr-7B, Neural-chat-7B, Phi-4-14B, Mistral-7B, and Llama-3.1-8B) also stand out as the top overall performers. Notably, Mistral-7B achieves the highest F1-score (86.0%)

Table 12.5: Performance of fine-tuning across different LLMs, showing average inference/training times and F1-score improvement.

Model	Avg. Inference Time (m:s)	Avg. Training Time (m:s)	Improved F1-score
zephyr-7B	01:07	33:31	+0.680
FinGPT v3.3-13B	01:03	27:32	+0.530
neural-chat-7B	01:07	33:48	+0.464
phi-4-14B	01:14	52:02	+0.415
Mistral-7B	00:37	33:44	+0.355
Llama-3.1-8B	00:40	28:47	+0.296
Qwen1.5-14B	01:21	49:33	+0.279
DeepSeek-R1-Distill-Qwen-7B	00:38	10:39	+0.077
Nous-Hermes-2-7B	00:40	19:12	+0.044
SOLAR-10.7B	00:44	27:32	+0.031
DeepSeek-R1-Distill-Llama-8B	00:40	11:23	+0.031
OLMo-7B	00:53	27:07	+0.016
Llama-2-13b	01:07	26:47	+0.008
gemma-7b	00:26	22:41	-0.001
QwQ-32B	02:33	63:18	-0.002
Starling-LM-7B	00:35	19:10	-0.004
Qwen2.5-7B	00:37	15:33	-0.038
gpt2-xl-1.6B	00:15	06:36	-0.047
Falcon3-10B	00:51	23:03	-0.056
Falcon3-7B	00:26	16:11	-0.063

while maintaining one of the fastest inference times (37 seconds) and requiring only a moderate fine-tuning time (33 minutes).

Interestingly, a few models (e.g., Falcon3-7B, QwQ-32B) perform worse than in the zero-shot scenario.

In summary, the analysis highlights that the benefits of fine-tuning vary considerably across models, and extended training does not always translate into improved performance.

Statistical Significance. We used McNemar's test [269] to determine the statistical significance of performance differences between zero-shot (ZS) and fine-tuned (FT) models on the same test set. This non-parametric test is designed for paired nominal data and is particularly suited for evaluating classifiers on the same test set and under the assumption of an existing better-than-random reference classifier [270, 271].

The complete results of the test are reported in Tab. 12.6. The test indicates that fine-tuning produced highly significant improvements ($p < 0.001$) for the three models that achieved the largest gains in F1 (Zephyr-7B, FinGPT v3.3-13B, and Neural-Chat-7B). In contrast, among the three models with the smallest improvements (gpt2-xl-1.6B, Falcon3-10B, and Falcon3-7B), only Falcon3-7B exhibited a statistically significant change ($p = 0.02075$). These findings suggest that fine-tuning consistently benefits models with stronger baseline performance, while its impact on weaker architectures is limited and often not statistically significant. Furthermore, comparing the best FT model (Mistral-7B) with the worst (Gemma-7B), McNemar's test ($\chi^2 = 888.338$, $p < 0.00001$) confirms that the performance difference between the two models is statistically significant, thereby reinforcing the conclusions drawn from the standard evaluation metrics.

Overall and Per-Class Metrics. Table 12.7 presents the comparison of accuracy, precision, recall, and F1-score, reported in both macro-averaged and weighted forms, before and after fine-tuning. As in the previous analysis, we focus on the six models that exhibit the highest and lowest performance gains during fine-tuning. The results indicate that the three models benefiting most from fine-tuning achieved consistent improvements across all reported metrics, with both precision and recall increasing simultaneously. In contrast, the models that exhibited performance degradation behaved more

Table 12.6: McNemar's test results Zero-Shot (ZS) vs Fine-Tuned (FT) models on the test set. Statistically significant results ($p < 0.05$) are highlighted in bold.

Best Three Models			*Worst Three Models*		
Model	Test Statistic	p-value	Model	Test Statistic	p-value
zephyr-7B	1123.905	**0.00000**	gpt2-xl-1.6B	3.285	0.06991
FinGPT v3.3-13B	811.133	**0.00000**	Falcon3-10B	2.766	0.09626
neural-chat-7B	906.368	**0.00000**	Falcon3-7B	5.348	**0.02075**

Table 12.7: Performance comparison of LLMs in ZS and FT. Metrics include accuracy, precision (PR), recall (RC), F1-score (F1), in both macro-averaged and weighted versions.

Model	Accuracy		Macro PR		Macro RC		Macro F1		Weighted-PR		Weighted-RC		Weighted-F1	
	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT
Models with the largest F1 improvement after fine-tuning.														
zephyr-7B	0.227	**0.840**	0.304	**0.823**	0.401	**0.819**	0.212	**0.821**	0.368	**0.840**	0.227	**0.840**	0.160	**0.840**
FinGPT v3.3-13B	0.096	**0.566**	0.365	**0.502**	0.334	**0.501**	0.059	**0.467**	0.549	**0.594**	0.096	**0.566**	0.018	**0.548**
neural-chat-7B	0.349	**0.848**	0.486	**0.824**	0.491	**0.799**	0.348	**0.811**	0.643	**0.847**	0.349	**0.848**	0.383	**0.847**
Models with the lowest F1 improvement after fine-tuning.														
gpt2-xl-1.6B	0.525	**0.539**	**0.414**	0.388	**0.374**	0.346	**0.352**	0.286	**0.519**	0.485	0.525	**0.539**	**0.472**	0.425
Falcon3-10B	**0.589**	0.575	0.479	**0.518**	**0.421**	0.389	**0.416**	0.360	0.560	**0.571**	**0.589**	0.575	**0.548**	0.493
Falcon3-7B	**0.582**	0.564	**0.479**	0.472	**0.422**	0.371	**0.417**	0.331	**0.565**	0.544	**0.582**	0.564	**0.538**	0.475

erratically, with some showing greater losses in recall and others in precision. For instance, the two Falcon models demonstrated a marked decline in recall after fine-tuning.

Table 12.8 reports the precision, recall, and F1 scores for the three classification categories: buy, sell, and other.

We observe several noteworthy patterns. For the models that achieve the largest F1 improvements after fine-tuning, recall consistently increases for the *buy* category and precision improves for the *sell* category, although this gain is partially counterbalanced by a decrease in recall for *sell*. In contrast, the *other* category shows consistent improvements across all metrics.

For the models that exhibited a performance decline after fine-tuning, the behavior differs considerably. They consistently show a reduction in the precision of the *buy* class, which is only partly compensated by an increase in recall. This indicates that the model becomes more prone to predicting the *buy* category even when it is not appropriate. At the same time, these models display a small but consistent improvement in the precision of the *sell* class. This improvement, however, is outweighed by a substantial reduction in recall, resulting in an overall decrease in the F1 score for the *sell* class. Finally, the *other* class is most strongly affected in terms of recall. GPT-2-xl and Falcon variants have

Table 12.8: ZS vs FT performance metrics for the best three and worst three models across the three classes. Best values per row are highlighted in bold.

Model	buy PR		buy RC		buy F1		sell PR		sell RC		sell F1		other PR		other RC		other F1	
	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT
Models with the largest F1 improvement after fine-tuning.																		
Zephyr-7B	0.479	**0.884**	0.020	**0.873**	0.039	**0.878**	0.178	**0.797**	**0.802**	0.777	0.292	**0.787**	0.254	**0.788**	0.382	**0.808**	0.305	**0.798**
FinGPT v3.3-13B	**1.000**	0.634	0.002	**0.781**	0.004	**0.700**	0.095	**0.247**	**1.000**	0.446	0.174	**0.318**	0.000	**0.625**	0.000	**0.278**	0.000	**0.385**
Neural-chat-7B	0.857	**0.880**	0.262	**0.897**	0.401	**0.888**	0.147	**0.775**	**0.866**	0.683	0.252	**0.726**	0.455	**0.817**	0.344	**0.818**	0.392	**0.817**
Models with the lowest F1 improvement after fine-tuning.																		
gpt2-xl-1.6B	**0.565**	0.548	0.834	**0.943**	0.673	**0.694**	0.124	**0.133**	**0.114**	0.020	**0.119**	0.035	**0.553**	0.483	**0.174**	0.075	**0.265**	0.130
Falcon3-10B	**0.605**	0.572	0.847	**0.936**	0.706	**0.710**	0.259	**0.355**	**0.074**	0.055	**0.115**	0.094	0.573	**0.627**	**0.341**	0.176	**0.428**	0.275
Falcon3-7B	**0.599**	0.564	0.867	**0.933**	**0.709**	0.703	0.240	**0.267**	**0.119**	0.020	**0.159**	0.037	**0.599**	0.585	**0.282**	0.160	**0.383**	0.251

Table 12.9: Comparison between the best and worst fine-tuned models in terms of classification performance.

Model	Accuracy	Macro Avg			Weighted Avg			Buy			Sell			Other		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Gemma-7B	0.366	0.459	0.427	0.323	0.570	0.366	0.386	0.480	0.560	0.520	0.190	0.250	0.220	0.210	0.270	0.240
Mistral-7B	0.860	0.842	0.829	0.835	0.861	0.860	0.860	0.880	0.920	0.900	0.770	0.800	0.780	0.840	0.820	0.830

relatively high recall in ZS for the *buy* class (≈0.83–0.87), but fine-tuning produces only marginal F1 improvements (e.g., GPT-2-xl buy F1 0.673 → 0.694) or even degradation in precision (e.g., Falcon3-7B *buy* precision 0.599 → 0.564).

Across all models, the *buy* category benefits the most from fine-tuning, especially in recall, while *sell* remains the most challenging class, with low precision and recall even for strong models. The *other* class shows intermediate behavior: strong models achieve substantial balanced improvements (e.g., Neural-chat-7B other F1 0.392 → 0.817), while weaker ones remain unstable.

The comparison between the best fine-tuned model (Mistral-7B) and the worst one (Gemma-7B) highlights the significant impact of fine-tuning quality on classification outcomes. As shown in Table 12.9, Mistral-7B achieves an overall accuracy of 86.0% and a macro F1-score of 0.835, whereas Gemma-7B performs substantially worse with an accuracy of only 36.6% and a macro F1-score of 0.323.

More interestingly, Mistral-7B demonstrates strong performance on the *buy* category, achieving an F1 score of 90%. In contrast, its performance on the *sell* category is comparatively weaker, with a precision of 77%. This discrepancy indicates that *buy* instances are more easily detected than *sell* instances, which constitutes a non-trivial phenomenon warranting further investigation. Interestingly, Gemma-7B also achieves the strongest performance on the *buy* class (F1 = 0.516), whereas its performance on the *other* class (F1 = 0.236) and the *sell* class (F1 = 0.218) is very weak.

RQ3 - Influence of Model Characteristics on Fine-Tuning Gains

To examine how the effectiveness of fine-tuning varies with the characteristics of the underlying model, Figure 12.2 reports the distribution of model performance, measured by the F1 score, across three dimensions: model family, training methodology, and release year.

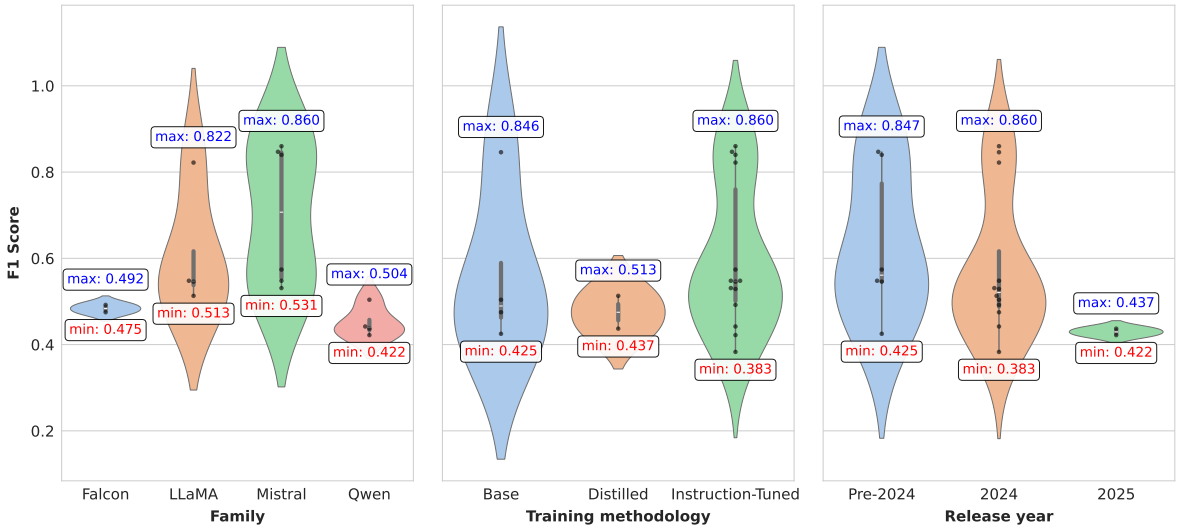


Figure 12.2: F1-score distribution of fine-tuned LLMs by model family, training methodology, and release year (only groups with at least two observations are included).

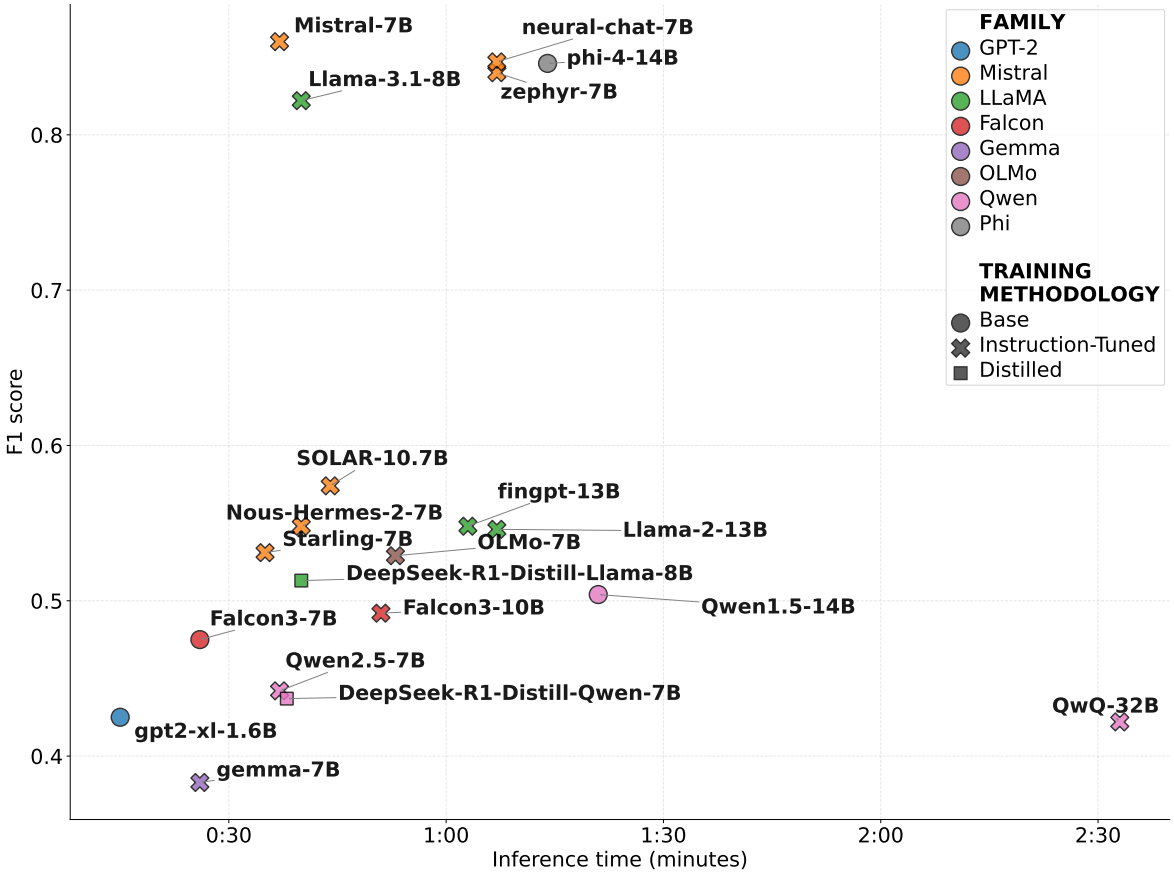


Figure 12.3: F1 scores of the fine-tuned LLMs as a function of their inference time. Marker color indicates the model family, marker shape represents the training methodology.

When comparing the different model families, the Mistral models achieve the best overall performance, reaching a maximum F1 score of 86.0% and consistently outperforming the other groups. In contrast, the LLaMA models exhibit lower accuracy; only one model reaches good performance. The Qwen and Falcon models perform less effectively for the considered task, yielding only limited results. If we consider the training methodology, instruction-tuning clearly emerges as the most effective approach. By contrast, base models obtain only moderate results, while distilled models perform the worst, exhibiting both the lowest minimum and maximum scores. These results indicate that, while distillation reduces model complexity, it simultaneously limits the model's capacity to generalize in this task. Finally, when considering the release year, models from 2024 and earlier achieve broadly similar results, with a slight advantage for the most recent ones. In contrast, models released in 2025 perform less well. However, this outcome may simply depend on the small sample size of 2025. Overall, the figure highlights two clear trends: the superiority of the Mistral family and the central role of instruction-tuning in boosting performance.

Figure 12.3 illustrates the relationship between F1 scores, inference time, model family, and training methodology. Among the five models with the highest F1-to-inference-time ratio, three belong to the Mistral family. Together with Llama-3.1-8B, these four models were also fine-tuned using instruction-based techniques. The plot further indicates that compact fine-tuned models (7–8B parameters) with relatively short inference times can match, and in some cases surpass, the performance of substantially larger models. In particular, Mistral-7B and Llama-3.1-8B distinguish themselves by combining strong predictive accuracy with highly efficient inference. These results suggest that such models are well-suited for powering classification frameworks that are both robust and scalable,

Table 12.10: Number of Type I errors (false positives) for each class in both configurations.

Model	buy		sell		other	
	FP(ZS)	FP(FT)	FP(ZS)	FP(FT)	FP(ZS)	FP(FT)
Models with the largest F1 improvement after fine-tuning.						
`Zephyr-7B`	25	132	746	40	871	168
`FinGPT v3.3-13B`	0	518	1918	275	1	129
`Neural-chat-7B`	50	141	1013	40	319	142
Models with the lowest F1 improvement after fine-tuning.						
`gpt2-xl-1.6B`	737	891	163	26	109	62
`Falcon3-10B`	633	802	43	20	197	81
`Falcon3-7B`	665	826	76	11	146	88

supporting the effective processing of the large volume of market-related posts generated daily.

RQ 4 - Error Types and Limitations

We conduct a comprehensive error analysis that includes a detailed breakdown of errors, such as classical Type I/II errors, domain-specific misclassifications (e.g., confusing a buy with a sell), and a qualitative assessment of their underlying causes (e.g., sarcasm or ambiguity). We then focus on the best-performing model (Mistral-7B) and examine its ability to generate coherent explanations for its own mistakes.

In line with the procedure adopted for RQ2, we first evaluate the effect of fine-tuning by comparing the three models that achieved the largest F1 improvements after fine-tuning (Zephyr-7B, FinGPT v3.3-13B, and Neural-Chat-7B) with the three models that exhibited the smallest, and in some cases negative, F1 gains (gpt2-xl-1.6B, Falcon3-10B, and Falcon3-7B). When compatible with the analysis, we also contrast the best-performing model overall (Mistral-7B) with the worst-performing one (Gemma-7B).

False positives and false negatives. In classification tasks, the analysis of Type I (false positives) and Type II (false negatives) errors offers a more nuanced understanding than overall accuracy alone. Examining these errors helps determine whether a model systematically overpredicts or underpredicts specific classes, thereby providing insights into class-wise precision and recall, uncovering potential biases, and assessing the effectiveness of fine-tuning strategies.

Table 12.10 reports the number of Type I errors (false positives) produced by the models under evaluation. For the models that showed performance gains after fine-tuning, the process significantly reduced false positives in the *sell* and *other* classes, at the cost of a limited increase in false positives for the *buy* class. For instance, Zephyr-7B decreases its false positives from 746 to 40 for the *sell* class and from 871 to 168 for the *other* class, highlighting a strong corrective effect of fine-tuning. FinGPT v3.3-13B, while initially showing a strong imbalance in the zero-shot setting with 1918 false positives in the *sell* class, achieves a much more balanced error distribution after fine-tuning (275 on *sell*, 518 on *buy*, and 129 on *other*).

The models that experienced a performance decline after fine-tuning displayed a consistent pattern. Specifically, false positives increased slightly for the *buy* classes, while they decreased marginally for the *sell* and *other* classes.

Overall, these results suggest that fine-tuning effectively mitigates error propagation in the better-performing models, while other architectures continue to exhibit a bias toward over-predicting the *buy*

Table 12.11: Number of Type II errors (false negatives) for each class in both configurations.

Model	buy		sell		other	
	FN(ZS)	FN(FT)	FN(ZS)	FN(FT)	FN(ZS)	FN(FT)
Models with the largest F1 improvement after fine-tuning.						
`Zephyr-7B`	1124	146	40	45	478	149
`FinGPT v3.3-13B`	1145	251	0	112	774	559
`Neural-chat-7B`	847	118	27	64	508	141
Models with the lowest F1 improvement after fine-tuning.						
`gpt2-xl-1.6B`	191	65	179	198	639	716
`Falcon3-10B`	176	74	187	191	510	638
`Falcon3-7B`	153	77	11	198	88	650

Table 12.12: Type I and Type II errors for fine-tuned Gemma-7B and Mistral-7B.

Model	Type I (FP)			Type II (FN)		
	Buy	Sell	Other	Buy	Sell	Other
`Gemma-7B`	379	910	56	616	66	663
`Mistral-7B`	114	36	146	121	49	126

class, indicating structural limitations in their representational capacity.

Table 12.11 reports the number of Type II errors (false negatives) produced by the evaluated models. The models that exhibited performance gains after fine-tuning show a substantial reduction in false negatives for the *buy* class and a moderate reduction for the *other* class. In contrast, the *sell* class experienced an increase in false negatives, although this effect was less pronounced for the best-performing models.

The models that do not benefit from fine-tuning still exhibit a reduction in false negatives for the *buy* class, but they suffer from an increased number of such errors in the *sell* and *other* classes.

Table 12.12 reports the distribution of Type I and Type II errors for the best and the worst fine-tuned models. The error patterns differ substantially between the two models. For Gemma-7B, false positives are dominated by the *sell* class (910), indicating a strong bias toward predicting *sell* even when incorrect. Moreover, false negatives are particularly high for both *buy* (616) and *other* (663), suggesting that the model often fails to detect these classes correctly. This imbalance reveals that Gemma struggles to establish a stable decision boundary across the three categories.

In contrast, Mistral-7B presents a more balanced error profile. Both false positives and false negatives remain relatively low across all classes. For instance, false positives for the *sell* class drop to only 36, compared to 910 in Gemma. Similarly, false negatives are reduced, especially for the *buy* class (121 vs. 616 in Gemma). These results indicate that fine-tuning enables Mistral to achieve a substantially better calibration, reducing both over-predictions and under-predictions.

Other Types of Errors. In addition to standard Type I and Type II errors, it is important to examine specific semantic misclassifications that are particularly relevant in the financial domain. We identify three types of errors that are especially critical.

The first is **action inversions**, which occur when the model predicts *sell* instead of *buy* (Buy $\rightarrow$ Sell) or *buy* instead of *sell* (Sell $\rightarrow$ Buy). These errors reverse the intended trading signals and can therefore be the most severe when the predictions are used for market forecasting or similar applica-

Table 12.13: Count of action misclassifications across models.

Model	Action Inversions				Non-action Misclassifications				False action misclassifications			
	Buy → Sell		Sell → Buy		Buy → Other		Sell → Other		Other → Buy		Other → Sell	
	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT	ZS	FT
Models with the largest F1 improvement after fine-tuning.												
Zephyr-7B	277	14	16	9	847	132	24	36	9	123	469	26
FinGPT v3.3-13B	1144	154	0	80	1	97	0	32	0	438	774	121
Neural-chat-7B	554	14	1	26	293	104	26	38	49	115	459	26
Models with the lowest F1 improvement after fine-tuning.												
gpt2-xl-1.6B	96	13	165	188	95	52	14	10	572	703	67	13
Falcon3-10B	24	11	142	173	152	63	45	18	491	629	19	9
Falcon3-7B	40	10	145	177	113	67	33	21	520	649	36	1

tions. The second is **non-action misclassifications** (Buy/Sell → Other), where actual trading signals are classified as non-actionable. Such errors lead to missed opportunities and can be particularly problematic when they cause weaker signals from social media or other sources to be overlooked. The third is **false action misclassifications** (Other → Buy/Sell), where non-actionable content is incorrectly classified as a trading signal. These errors introduce spurious signals, increase noise in the data, and may provoke misleading indications that confound subsequent applications relying on the predictions.

Analyzing these errors highlights the practical limitations of language models in financial decision-making and emphasizes the risks associated with misaligned predictions.

Table 12.13 presents the error types for the six models that exhibit the largest and smallest F1 improvements after fine-tuning.

Overall, the three models that benefit the most from fine-tuning show a substantial reduction in action inversion errors, which is highly positive given that these represent the most severe type of mistake. In addition, non-action misclassifications also tend to decrease, indicating that the fine-tuned models are more sensitive to distinguishing between buy and sell decisions rather than defaulting to other when uncertain. Nevertheless, the behavior of false action misclassifications remains more nuanced. While the frequency of *Other* → *Sell* errors decreases after fine-tuning, *Other* → *Buy* errors increase, suggesting that the model can sometimes become overly confident in predicting *buy*.

Notably, Zephyr shows relatively low action inversion counts after fine-tuning (14 and 9), while FinGPT has extremely high Buy → Sell errors in zero-shot (1144) but drastically improves after fine-tuning (14), demonstrating the impact of task-specific adaptation. Zephyr also maintains moderate counts in non-action misclassifications category, whereas Neural-chat shows a reduction after fine-tuning, reflecting improved sensitivity to actionable content. For the false action misclassifications, fine-tuning consistently reduces these errors across best models; for example, Zephyr's Other → Buy decreases from 9 to 123, and Other → Sell from 469 to 26, highlighting that fine-tuning enhances both precision and reliability of trading signal predictions. In contrast, the worst-performing models exhibit higher counts across most error types, with minimal improvements from fine-tuning, emphasizing their limitations in financial text classification tasks.

Table 12.14 summarises the distribution of semantic errors for the best and worst fine-tuned models.

Gemma-7B is dominated by inversion errors, with 574 cases of misclassifying *buy* as *sell*, a critical type of error in financial applications since it represents a direct reversal of the intended action. Additionally, it produces a large number of false actions from the *other* class (327 as *buy*, 336 as *sell*), highlighting a tendency to over-predict financial actions in neutral contexts. Conversely, non-action errors for actual *buy* or *sell* cases are relatively limited (42 and 14, respectively).

Mistral-7B rarely suffers from inversion errors (16 and 8), suggesting that the model is much more reliable in distinguishing opposite actions. However, it shows higher vulnerability to non-action

Table 12.14: Semantic error analysis for fine-tuned Gemma-7B and Mistral-7B.

Error Category	Error Type	Gemma-7B	Mistral-7B
Action Inversions	Buy→Sell	579	16
	Sell→Buy	52	8
Non-action Misclassifications	Buy→Other	42	105
	Sell→Other	14	41
False Action Misclassifications	Other→Buy	327	106
	Other→Sell	336	20

misclassifications in the opposite direction: 105 cases of *buy→other* and 41 cases of *sell→other*, indicating a tendency to underestimate actionable decisions. Errors from *other→buy*/*sell* remain moderate compared to Gemma (106 vs 20).

Corrections and New Errors Introduced by Fine-Tuning. It is crucial to analyze not only the errors corrected relative to the zero-shot setting but also the new errors introduced. Such a comparison offers a more comprehensive view of the model's behavioral changes, revealing which types of predictions benefit most from fine-tuning and where trade-offs may occur.

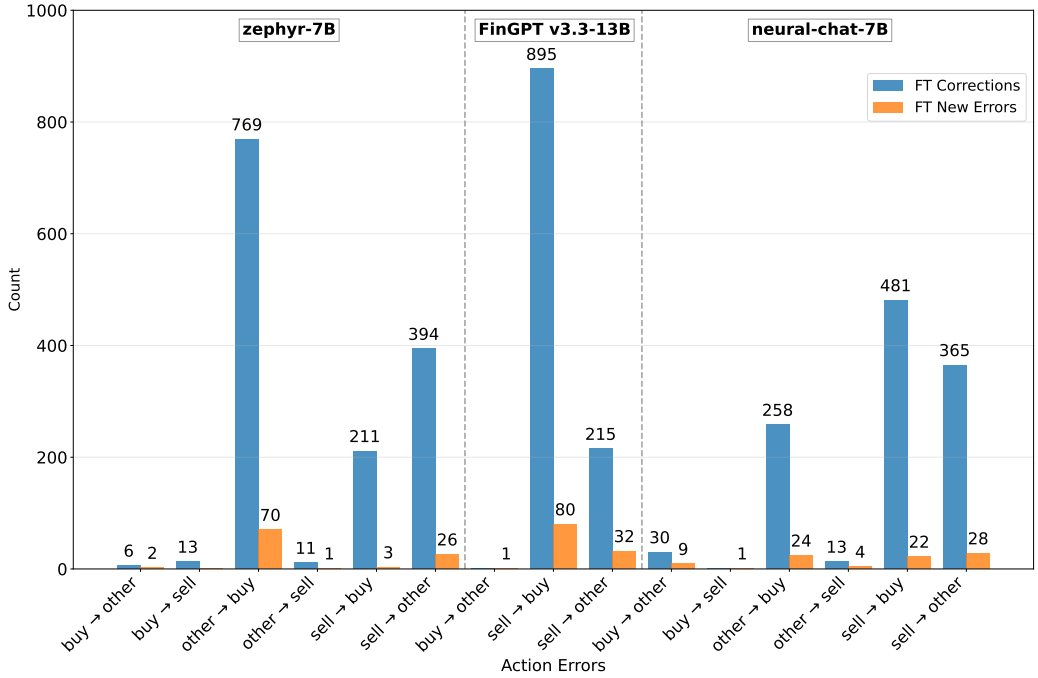


Figure 12.4: Corrections and New Errors Introduced by Fine-Tuning in the models with the largest F1 improvement after fine-tuning.

Figure 12.4 and Figure 12.5 show the distribution of semantic errors that were corrected (blue) and those that were introduced (orange) after fine-tuning. The results indicate that the types of errors corrected or introduced are strongly influenced by the characteristics of each model. A common pattern is the correction of *sell→other* errors in the model that benefited the most from the process, and the introduction of new *other→buy* errors in the model that did not benefit from fine-tuning.

For zephyr-7B, fine-tuning corrected a total of 1,404 errors, primarily involving misclassifications

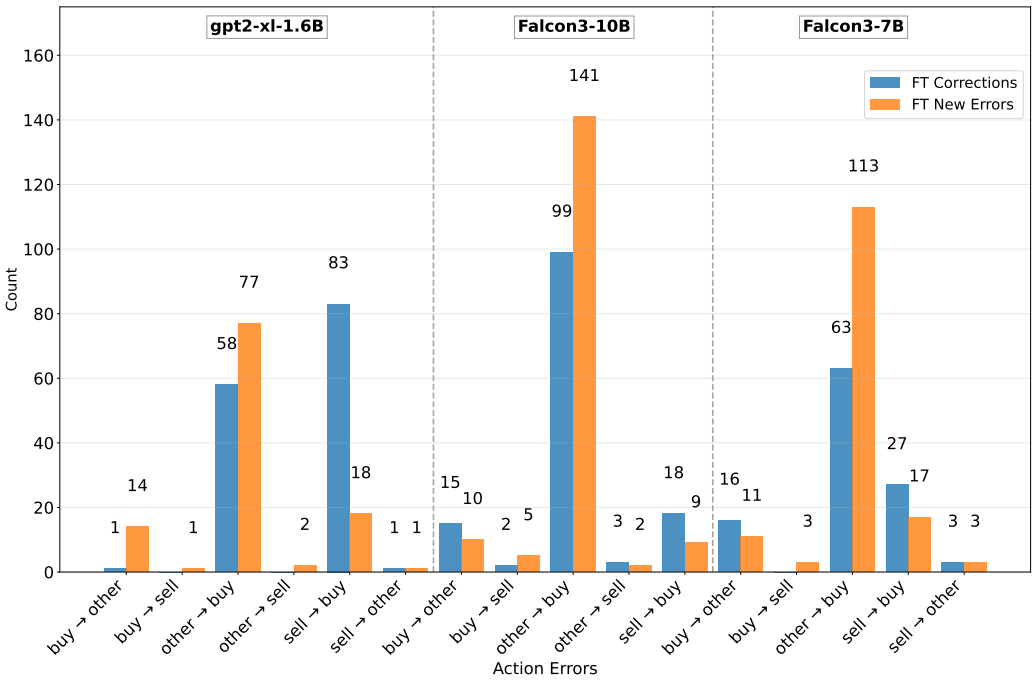


Figure 12.5: Corrections and New Errors Introduced by Fine-Tuning in the models with the lowest F1 improvement after fine-tuning.

of *buy* instances predicted as *other* (769 cases) or *sell* (211 cases), as well as *other* instances incorrectly predicted as actions in the zero-shot setting. At the same time, 102 new errors were introduced, mostly due to *other* instances being misclassified as *buy* (70 cases).

The FinGPT v3.3-13B model corrected 1,110 errors, with the largest share (895 cases) involving *buy* instances that were previously mislabeled as *sell*. However, this improvement came with 113 new errors, notably the frequent confusion of *sell* instances with the *buy* class (80 cases) or the *other* class (32 cases).

Finally, neural-chat-7B achieved 1,147 error corrections, most of which were *buy* instances previously predicted as *sell* (481 cases) or *other* (258 cases). This model also corrected a considerable number of *other* instances misclassified as actions (395 cases). The number of errors introduced was limited to 88, with the most frequent being *sell* instances predicted as either *buy* (22 cases) or *other* (28 cases).

For Falcon3-7B, fine-tuning corrected 109 errors, mostly involving misclassified *buy* samples that were previously predicted as *other* or *sell*. However, the model introduced 147 new errors, with the majority related to the *other* class being incorrectly predicted as *buy*. This suggests that fine-tuning led to an over-adjustment, with the model over-predicting *buy* instances.

Falcon3-10B exhibits a similar trend. While 137 errors were corrected, primarily involving *buy* instances misclassified as *other* or *sell*, fine-tuning introduced 167 new errors. Once again, the dominant issue lies in the *other* class being incorrectly labeled as *buy*, indicating that fine-tuning shifted the model's decision boundary excessively towards the *buy* class.

Finally, gpt2-xl-1.6B corrected 143 errors, largely improving the recognition of *buy* cases that were previously predicted as *sell*. Nevertheless, 113 new errors were introduced, with the majority again involving *other* being misclassified as *buy*. Although the number of corrections slightly outweighs the errors introduced, the imbalance among classes remains problematic.

Overall, it appears that the last three models, which were unable to benefit from fine-tuning, exhibited particular difficulties in misclassifying the other category, which was often incorrectly labeled

as buy. This may be due to the tendency of the models to conflate a general, potentially positive sentiment toward a stock with an explicit indication to trade.

Qualitative Analysis of Underlying Error Causes. In order to gain a deeper understanding of the limitations of large language models in financial classification tasks, we conducted a qualitative analysis of their errors. We began with a manual examination of misclassified posts from various models, which allowed us to identify recurring patterns and linguistic challenges. From this analysis, we drafted a preliminary taxonomy of six error categories.

To validate this initial set, we employed OpenAI's GPT-5, prompting it to evaluate the distinctiveness and conceptual overlap of the proposed categories. The model suggested consolidating several categories, resulting in a refined taxonomy of four distinct error categories. This hybrid approach ensures that our final error taxonomy is not only grounded in empirical observation but also optimized for conceptual rigor.

The four error categories identified are the following:

- **Category 1: Ambiguity and Implicit Language:** Errors arising from vague, indirect, or context-dependent statements, where the intended action is not explicitly stated.
- **Category 2: Generic Opinions vs. Actual Actions:** Misclassifications between texts expressing general opinions or sentiments and those describing concrete trading actions.
- **Category 3: Noise, Memes, Irony, and Sarcasm:** Errors due to informal or humorous language, often including memes, exaggerations, or irony that obscure the underlying meaning.
- **Category 4: Informational or News-Like Content:** Cases where factual reporting of news or events is mistaken for an actionable trading signal.

Each error category reflects a different type of linguistic or semantic challenge: for example, ambiguous or ironic texts tend to be harder to map to clear actions, while news-like content may mislead the models into predicting *buy* or *sell* when no explicit action is suggested. This allows us to better understand whether the errors are due to fundamental limitations in semantic comprehension, sensitivity to noise, or the inability to handle implicit or contradictory information correctly.

Table 12.15: Error distribution across categories for different LLMs.

Model	Ambiguity and Implicit Language	Generic Opinions vs. Actual Actions	Informational or News-Like Content	Noise, Memes, Irony, and Sarcasm
Best three models:				
Zephyr-7B	271	7	20	42
FinGPT v3.3-13B	774	19	39	88
Neural-Chat-7B	266	6	18	33
Worst three models:				
GPT-2-XL-1.6B	823	24	42	88
Falcon3-10B	758	23	43	76
Falcon3-7B	786	23	41	72
Overall Worst: Gemma-7B	1017	24	41	260
Overall Best: Mistral-7B	228	7	21	41

Ambiguity and Implicit Language. overwhelmingly dominates the error space, representing the most frequent source of misclassification across all models. This confirms that vague or context-dependent financial texts remain the hardest to interpret correctly. To address these cases, it may be beneficial to incorporate additional contextual information, for instance from surrounding posts

or user profiles. *Noise, Memes, Irony, and Sarcasm* represent the second major source of errors and remain difficult to handle even for the strongest models. Interestingly, the particularly high number of such errors in Gemma-7B indicates that this is one of the primary points of failure for models that are weaker on this task. *Informational or News-Like Content* also remains problematic, although the best-performing model shows a clear improvement in handling these cases. Finally, errors related to *Generic Opinions vs. Actual Actions* are comparatively rare, especially in the best model. This suggests that fine-tuned models are able to discriminate effectively between generic statements and personal intentions or suggestions regarding trade actions.

Analysis of the Model Explanations. We evaluated the explanation capabilities of the best-performing model, fine-tuned Mistral-7B, by instructing it to provide natural language explanations for its predictions. This analysis distinguishes correct predictions with coherent reasoning from potentially fortuitous ones, while revealing failure modes such as handling sarcasm, missing subtle cues, or confusing negativity with sell intent.

The error analysis identifies three persistent challenges: (i) *Over-sensitivity to optimism*, where general positive sentiment is incorrectly interpreted as a specific *buy* recommendation; (ii) *Sequential action confusion*, in which attention to initial verbs overlooks final trading intentions (e.g., "sold X for buying Y"), leading to label inversion; and (iii) *Informality and irony*, where the model fails to detect the underlying intent in posts containing slang or humorous expressions.

These patterns emerged consistently across various error cases and underscore the model's difficulty in interpreting community-specific communication patterns, understanding the strategic intent embedded in transactional language, and accounting for emotional cues that influence directive meaning. To illustrate these challenges concretely, we present six representative error cases in the blue boxes, which exemplify the most significant and recurring failure modes observed in our analysis.

In the positive cases, the model demonstrates nuanced understanding of financial sentiment. It correctly identifies urgent selling pressure through bearish terminology, recognizes non-actionable financial discussion, and decodes informal trading jargon about buying strategies. These patterns indicate the model has learned meaningful distinctions between financial expression types.

Collectively, these errors highlight the model's challenges in deciphering community-specific communication patterns, the strategic intent behind transactional language, and the emotional modifiers that modify directive force. Future improvements should focus on enhancing contextual reasoning and financial discourse pragmatics.

12.4 Discussion

In the following sections, we discuss the main results obtained from the experiments, explain how they address the research questions, and outline their theoretical and practical implications.

RQ1 - How effective are modern LLMs in performing STAD compared with traditional models, and what is the best-performing LLM Model?

The experiments demonstrate that LLMs can outperform traditional models in STAD, but their effectiveness critically depends on fine-tuning. While simple baselines such as LSTM and Bi-LSTM do not perform well, an MLP combined with high-quality embeddings sets a strong non-LLM baseline, surpassing all zero-shot LLMs. Among the latter, Gemma-3-27B delivers the best zero-shot performance. By contrast, fine-tuned LLMs, led by Mistral-7B with an F1 of 86.0%, achieve substantial gains, clearly outperforming both traditional models and zero-shot LLMs.

The weak performance of FinGPT, a language model specifically optimized for financial sentiment analysis, supports one of the main hypotheses discussed in the introduction: sentiment analysis methods, even when tailored to the financial domain, are not well suited to address STAD. Although

Positive Prediction with Explanation

Example 1: *DUMP IT BEFORE IT CRASHES*
Real Label: sell
Predicted Label: sell
Model Explanation: The financial sentiment expressed in the post is negative, indicating a desire to sell an asset. The use of the word "crashes" further emphasizes this sentiment, suggesting a belief that the asset's value is about to decrease significantly. Therefore, the label assigned to this post is "sell."

Example 2: *Just a reminder of what could have been*
Real Label: other
Predicted Label: other
Model Explanation: The given post does not express any clear financial sentiment. It appears to be a reflective statement about a missed opportunity rather than an expression of optimism or pessimism regarding the financial markets. Therefore, it falls under the "other" category in financial sentiment classification.

Example 3: *Lowest dip it'll hit today. When's a good time buy a lot I just put into AMC at a pop*
Real Label: buy
Predicted Label: buy
Model Explanation: The financial sentiment expressed in this post is bullish, indicating a positive outlook towards the financial markets or a particular asset. The user is expressing their intention to buy more of a specific stock (AMC) and is optimistic about its future price movement. This sentiment can be classified as "buy" or "bullish."

Negative Prediction with Explanation

Example 1: *were cool guys this whole week was a huge step forward Itll get even better next week because we will hold together*
Real Label: other
Predicted Label: buy
Model Explanation: The financial sentiment expressed in the post is positive and optimistic. The author expresses confidence in the market and their investment decisions, using phrases like "huge step forward" and "it'll get even better next week." This sentiment is consistent with a buy signal, as it indicates a bullish outlook and a willingness to hold onto investments.

Example 2: *sold all my bag holding weed stocks for another gme*
Real Label: buy
Predicted Label: sell
Model Explanation: The financial sentiment expressed in this post is one of selling, likely due to the author's decision to sell their holdings in weed stocks and invest in GameStop (GME) instead. This sentiment is often associated with a bearish outlook on the market or specific securities, as the author is reducing their exposure to certain assets.

Example 3: *yes now TRY TO SELL IT LOLOL*
Real Label: sell
Predicted Label: other
Model Explanation: The given text "TRY TO SELL IT LOLOL" does not express a clear financial sentiment. The use of the word "sell" could be interpreted as a bearish sentiment, but the inclusion of "LOLOL" suggests that the author may be joking or expressing excitement rather than fear or pessimism. Without additional context, it is difficult to definitively classify this text as bullish, bearish, or neutral. Therefore, it is labeled as "other" to reflect its ambiguous nature.

sentiment analysis is often used as a proxy for this task [253, 254, 13], our results suggest that it fails to capture the underlying structure of action-oriented discourse. This indicates the need to focus more on detecting the actions implied or suggested by the text rather than on assessing its general sentiment. It also highlights the importance of developing alternative techniques for exploring this space that cannot rely on off-the-shelf sentiment analysis tools. This work represents an important first step in this direction, introducing a high-quality dataset and providing a detailed analysis of the behaviour

of LLMs in this context.

Furthermore, these findings have several additional implications for the effectiveness of current NLP methods on STAD: 1) zero-shot LLMs are not inherently superior to well-designed traditional pipelines for this task; 2) fine-tuning is essential to fully unlock their potential; and 3) parameter-efficient approaches, such as LoRA, make such adaptation practically feasible.

RQ2 - Does fine-tuning LLMs on high-quality data significantly enhance their performance on this task?

The experiments provide clear evidence that fine-tuning on high-quality data can significantly enhance performance, but with strong variation across model families. On average, fine-tuning yielded an F1 increase of over 15%, with several modern, instruction-tuned 7–8B models (e.g., Mistral-7B, Zephyr-7B, Neural-Chat-7B) achieving improvements above 25% and reaching competitive accuracy levels with relatively efficient inference times. Statistical testing confirmed that these gains were highly significant. Conversely, a few models showed marginal or even negative effects, indicating that fine-tuning is not uniformly beneficial and may exacerbate weaknesses in less capable architectures.

These findings have important practical implications, as many companies in this domain often specialize existing models with the expectation that fine-tuning will enhance performance relative to the base version. Our results in this field are instead consistent with a few prior studies showing that fine-tuning does not always lead to substantial performance improvements [249, 250, 251]. Overall, the evidence suggests that fine-tuning can be a powerful approach, particularly when applied to instruction-tuned models; however, its effectiveness strongly depends on the underlying model architecture, as discussed in the following sections.

RQ3 - To what extent does the effectiveness of fine-tuning depend on the characteristics of the underlying model?

The experiments show that the effectiveness of fine-tuning is strongly shaped by the characteristics of the underlying model. Model family plays a central role, with Mistral models consistently outperforming others and demonstrating the best balance between accuracy and efficiency. Training methodology is equally critical: instruction-tuned models yield markedly higher gains than base or distilled variants. Finally, efficiency analyses reveal that smaller models such as Mistral-7B and LLaMA-3.1-8B can rival or exceed much larger counterparts.

Overall, these findings offer a comprehensive overview of the current state of the art for this challenging and relatively unexplored task, while also identifying an effective baseline configuration that can be directly adopted by organizations aiming to detect user actions in this domain. The recommended setup is based on open-weight, instruction-tuned language models of moderate size (7–8B parameters), drawn from the most suitable model families (e.g., Mistral, LLaMA). When fine-tuned on our domain-specific datasets, these models demonstrate strong and consistent performance, achieving a favorable balance between accuracy and computational efficiency. Furthermore, their open-weight nature facilitates transparency, reproducibility, and ease of integration into existing industrial pipelines.

RQ4 - What types of errors occur most frequently, and how does fine-tuning influence their distribution, severity, and nature?

The error analysis shows that, prior to fine-tuning, models are dominated by ambiguity and implicit-language failures, with irony and sarcasm representing the second major source of errors. In the zero-shot setting, weaker models also tend to over-predict *buy*, generating both false actions (Other $\rightarrow$ Buy/Sell) and harmful action inversions.

Fine-tuning recalibrates the stronger models by substantially reducing false positives on sell/other, lowering false negatives on buy, and decreasing both inversion and false-action errors. The remaining mistakes are mostly shifted toward non-action misclassifications (buy/sell $\rightarrow$ other), which are safer

because they avoid trades rather than triggering incorrect ones. By contrast, models that do not benefit from fine-tuning exhibit an increase in buy-biased false positives and introduce other $\rightarrow$ buy errors, indicating an over-correction rather than an improved understanding.

The qualitative analysis confirms that ambiguity and implicit language remain the leading source of misclassification, underscoring the persistent difficulty of interpreting vague or context-dependent financial texts. Noise, memes, irony, and sarcasm form the second main error category, particularly affecting weaker models such as Gemma-7B. Although informational or news-like content remains challenging, stronger models achieve significant improvements, and errors involving generic opinions versus explicit actions are comparatively rare.

The main implication of these findings is the need for further research into the ability of LLMs to navigate and understand complex online language. Future work should also investigate more sophisticated processing pipelines that combine LLMs with additional contextual information extracted from relevant online communities. Such integration, which we plan to explore in future work, could enhance the models' capacity to interpret nuanced discourse patterns, social dynamics, and domain-specific conventions that characterise these environments.

Finally, the analysis of Mistral-7B explanations indicates that requiring models to generate justifications can provide valuable insights into their reasoning in this domain. Specifically, we observed cases where the model misclassified optimism as buy signals, confused sequential trading actions, or failed with irony and slang. These errors reflect an occasional reliance on surface cues rather than deeper pragmatic reasoning. Such qualitative analyses can also serve as a practical tool for stakeholders to quickly assess a model's understanding of complex and noisy online environments.

Overall, our experiments reveal persistent weaknesses in the current generation of language models when dealing with text types that demand extensive contextual comprehension and discourse-level interpretation. In particular, the ability to analyze a conversation within the broader community context, and to understand how collective dynamics, memes, and irony shape online discourse, remains limited.

The analysis presented in this chapter represents an initial step in this direction. Future work should further investigate these aspects across multiple domains, exploring methods to enhance contextual grounding, pragmatic reasoning, and the robustness of model explanations in diverse social media environments.

12.5 Conclusion

Since online information exchange has been shown to influence market trends [104], monitoring the behaviour of retail investors has become increasingly important [110, 16]. Theoretical models that account for the role of social networks in shaping asset prices require reliable indicators of investors' online intentions toward specific stocks. However, traditional financial sentiment analysis techniques are limited in their ability to capture explicit trading intentions or direct recommendations to buy or sell individual stocks.

This work explores how LLMs can be employed to address challenges in financial NLP tasks, with particular attention to the role of fine-tuning and to the common limitations and errors that arise. To this end, we introduce a new task, *STAD* (Social Trading Action Detection), which aims to infer users' trading intentions by classifying online posts into three categories: *buy*, *sell*, or *other*. We also release FINREDDIT-2K, a dataset of 2,123 manually annotated Reddit posts. We then conduct a systematic study on the effects of fine-tuning across a range of LLMs. In addition to identifying the best-performing architectures, we analyse how fine-tuning interacts with different model characteristics and the extent to which it mitigates or exacerbates specific types of errors.

Our findings, structured around four research questions, indicate that LLMs provide significant

advantages over alternative approaches, particularly when fine-tuned for the target task. The experiments show that modern LLMs can substantially outperform traditional models in STAD, but only when trained on high-quality data. In contrast, zero-shot LLMs underperform compared to a strong MLP baseline, confirming that fine-tuning is essential to fully exploit their potential. The performance gains, however, vary considerably across architectures. Instruction-tuned mid-sized models (e.g., Mistral-7B) achieve the best balance between accuracy and efficiency, while some models may fail to benefit and can even degrade after fine-tuning. The error analysis further reveals that fine-tuning consistently reduces severe mistakes such as action inversions and biased false positives, shifting the error distribution toward safer misclassifications. Nonetheless, ambiguity, irony, and implicit language remain persistent challenges, underscoring the need for more advanced contextual reasoning. Future studies should therefore explore in greater depth how conversations evolve within broader community contexts and how to enhance the understanding of texts shaped by collective dynamics and ironic expression. Overall, our results demonstrate that effective fine-tuning strategies, combined with robust model families, are crucial for attaining state-of-the-art performance in this challenging task.

Future research will proceed along two main directions. The first involves investigating state-of-the-art multimodal models that can process images and videos, which are frequently present in relevant online posts and may provide essential cues for a more accurate understanding of their content. The second focuses on integrating the results of the STAD task into asset price prediction frameworks with the goal of improving the reliability of stock price movement forecasts.

Chapter 13

Enhancing user reliability using a contextual-based approach on heterogeneous graph

The proliferation of social media platforms has fundamentally transformed information dynamics within financial markets, establishing online communities as critical venues for sentiment dissemination and collective decision-making. Platforms like Reddit now serve as real-time barometers of market psychology, where user-generated content directly influences investment behaviors and can precipitate herd-like market movements. Despite this recognized impact, the systematic analysis of social data for financial applications remains challenging, primarily due to the scale of digital interactions and the complex, multi-dimensional nature of influence in financial discourse.

The work presented in this chapter addresses these challenges by developing a comprehensive framework that integrates advanced social network analysis with sophisticated NLP techniques to quantify user influence in financial contexts. We introduce the *Content-based Centrality* (CbC) score [6], a novel metric that synthesizes user engagement patterns, sentiment dynamics derived from post content, and structural network position. Through extensive evaluation on real-world data from Reddit's financial communities, we demonstrate that the *CbC* score significantly outperforms traditional centrality measures in identifying financially relevant influence.

Our work makes three principal contributions to the emerging field of social-driven finance: first, we establish a robust methodological framework for extracting financial signals from social media data; second, we validate our approach through rigorous human evaluation, assessing both the relevance and trustworthiness of identified influential content; third, we provide empirical evidence of how digital social interactions translate into measurable financial influence. By bridging the gap between traditional financial analysis and computational social science, this research offers valuable insights for monitoring market sentiment, identifying emerging trends, and understanding the mechanisms through which online communities shape financial market behaviors.

13.1 Introduction

In recent years, the rise of social media platforms (e.g., Reddit[1], StockTwits[2]) in shaping market dynamics affects the effectiveness of Artificial Intelligence models in analyzing market behavior [272].

Reddit discussions are organized into specialized communities known as *subreddits*. Each subreddit treats a specific theme or subject. Within each subreddit, users can create *submissions*, which serve

[1] https://www.reddit.com/
[2] https://stocktwits.com/

as the primary content that others can comment on or engage with. Submissions typically contain text, links, or media, and each submission can receive *upvotes* or *downvotes* based on how useful or engaging the community finds it. Reddit's structure is inherently hierarchical. Each submission can attract comments, generating conversation threads as users reply to one another. This dynamic allows for in-depth, community-driven discussions where users can share opinions, insights, and analysis, making Reddit a rich source of organic, user-generated data [116]. One key aspect of the platform is the *karma* system, where users accumulate points based on the reception of their posts and comments. Karma serves as a form of social reputation and is a proxy for measuring users' perceived influence or trustworthiness within the community.

In financial subreddits, discussions often revolve around stock performance, market trends, and investment strategies, making Reddit a valuable resource for financial analysis [273, 126]. These financial communities have become a breeding ground for retail investors to exchange ideas, speculate on stock movements, and, in some cases, drive market behavior through collective actions [116]. The *meme stock* phenomenon, which brought unprecedented attention to stocks like GameStop (GME) and AMC, is a prominent example of how sentiment generated on Reddit can have real-world financial impact [274].

We focus on finance-related subreddits: *wallstreetbets*, *finance*, *economics*, *investing*, *pennystocks*, *StockMarket*, *Stocks*, and *Dividends*. These subreddits were selected due to their central role in financial discussions on Reddit, where users actively debate market trends, stock analysis, and investment strategies. Their influence on real-world market movements, as seen with events like the GameStop short squeeze, makes them highly relevant for studying user impact and opinion leadership in online financial communities.

User communities on Reddit have influenced stock prices through cooperative actions, as seen in cases like GameStop (GME) and AMC [274]. Nevertheless, identifying relevant users has become more and more challenging [275, 276] since it is necessary to investigate both textual content and dynamic interaction among users [122]. Various studies focus on modeling these interactions by quantifying the strength of user connections based on interaction frequency or exploiting the herd effect, where users follow those perceived as influential leaders [120, 121]. However, evaluating user trust within a social media community necessitates a thorough investigation of both the content and nature of interactions, which requires substantial contextual information [277, 278]. Specifically, user-generated content frequently captures the attention of others and significantly influences collective behavior and thoughts [116, 279]. Hence, integrating content-based analyses with network topology measures is crucial for a more comprehensive understanding of user influence.

In this work, we aim to investigate the user influence by designing a novel index called *Content-based Centrality score* (CbC). It incorporates both a user centrality measure within the community (*Centrality-based Influence Score* (CIS)) and a content-based component (*Sentiment Engagement Score (SEI)*), which considers the sentiment of a post and the level of engagement it receives from other users. We evaluated our methodology on a real-world dataset consisting of over $68,084$ posts from prominent financial Reddit communities discussing a set of 50 stocks. To obtain a structured view of interactions within the Reddit financial ecosystem, we built a graph-based representation of subreddit communities composed of users, posts, and discussed stocks. This approach enables us to apply network-based metrics to identify the most influential users. Additionally, we extract sentiment from post content and use it as a weighting factor to assess community engagement, reflected in the upvotes and downvotes each post receives.

An experimental evaluation has been performed on 8 subreddits considering a total of $14,947$ submissions to identify more than $119,000$ nodes and $205,000$ edges. On top of this graph, we compare the proposed methodology w.r.t. several baselines, also providing a qualitative evaluation made by human evaluators.

The main contributions of this work can be summarized as follows:

- Designing a heterogeneous graph-based model that mirrors the interactions in the Reddit communities.
- Defining a metric to measure a user's influence accounting for users' interactions within the social media network and the engagement related to the sentiment of the posts they share.
- Bridging the gap between traditional finance and digital social interaction to provide a deeper understanding of the "herding effect" generated by online communities.

The following sections will detail the data collection and analysis methodology, feature selection and graph construction criteria, and the centrality and sentiment metrics adopted to identify influential users. Finally, we will discuss the results and potential implications for studying social dynamics in financial markets.

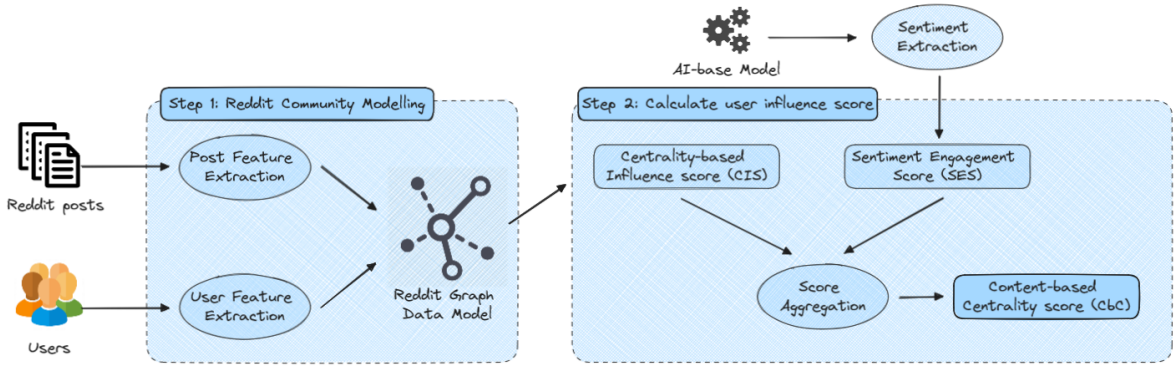


Figure 13.1: The workflow outlines the process of modeling the Reddit community as an heterogeneous graph, extracting post and user features to build a *Centrality-based influence score* (CIS). Post sentiment is used to construct the *Sentiment engagement score* (SES). These metrics are then combined to create a *Content-based centrality score* (CbC), as detailed in the subsequent sections.

13.2 Methodology

In this section, we detail the methodology employed to analyze user influence within the Reddit community. We begin by describing the process of constructing the graph, including users, posts, and submissions, along with their interactions. Subsequently, we introduce the new context-based metric for quantifying user influence, integrating centrality measures derived from the graph with sentiment analysis. This approach enhances our understanding of user engagement and its impact on the community. As illustrated in Figure 13.1, the methodology presented in this study consists of two main stages, which integrate structural and content-based elements, providing a comprehensive approach to evaluating a user's impact on the platform.

1. *Step 1: Reddit Community Modelling*. Initial phase, where we construct the Reddit graph by extracting features from posts and users components(detailed in Section 13.2.1).
2. *Step 2: Calculating User Influence Score*. Building upon the graph model, we compute the user influence score considering both centrality within the network and the sentiment extracted from the posts. We provide the details regarding constructing the three scores in Section 13.2.2.

The final task of this study involves developing a methodology to assign a relevance index to each Reddit user based on their posts and user-related information. By analyzing user activity and interactions within the Reddit community, as well as the sentiment and engagement of their posts,

we calculate an influence score that captures the user's impact on the platform. This approach uses centrality measures and sentiment analysis to provide a comprehensive view of each user's relevance, enabling the identification of key influencers.

13.2.1 Graph data model

Starting from the list of subreddits and stocks, we gathered various data related to submissions and their associated comments. Specifically, for each submission and its corresponding comments, we collected the text, publication date, total number of comments, and the number of upvotes and downvotes. Additionally, we extracted information regarding each user's account creation date and karma score.

The primary entities of interest are subreddits, stocks, users, submissions, posts, and replies. These elements form the building blocks of our interaction graph, allowing us to model the relationships between users and their influence on financial discussions.

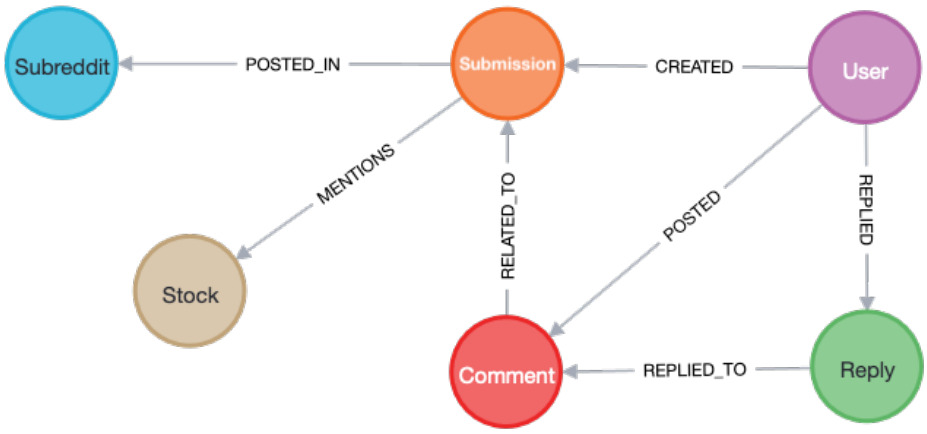


Figure 13.2: Graph model of the entities involved and their connections

Figure 13.2 illustrates the graph structure with the interaction between entities in the Reddit community, represented as nodes with distinct features. The *Subreddit* nodes represent eight communities selected for our analysis, each characterized by its name and the number of members. *Submission* nodes capture individual posts, identified by their title, posting date, total number of comments, upvotes, and downvotes. *Comment* nodes represent user replies to a submission, including features such as posting date, total number of replies, upvotes, and downvotes. Similarly, *Reply* nodes represent users' replies on comments, including posting date, score, upvotes, and downvotes. The *User* nodes link each post (submission, comment, reply) to the user who posted it. We further describe each user with their username, total karma, and active period in the community, calculated from the date of their first submission up to the latest date in the dataset (08/07/2024). Lastly, the *Stock* nodes represent stocks mentioned in each submission, establishing a link between discussions on Reddit and real-world financial assets.

13.2.2 Content-based Centrality score

To quantify the influence of users within the Reddit financial community, we propose the **Content-based Centrality score** (CbC), a metric that combines two complementary dimensions. The first component, **Centrality-based Influence Score** (CIS), focuses on the structural position of users within the interaction graph. Higher centrality scores indicate users who play a pivotal role in spreading information or shaping discussions.

The second component, **Sentiment Engagement Score** (SES), shifts the focus from network structure to content. It assesses the tone and sentiment of the posts made by each user about specific stocks. By conducting sentiment analysis on user-generated content, we measure their posts' positive, neutral, or negative stance and evaluate how these sentiments potentially influence market behavior. This dual approach enables us to capture Reddit's social dynamics and the financial sentiment users generate through their discussions.

Centrality-based Influence Score (CIS) To ensure a comprehensive assessment of user influence, we use PageRank as the centrality metric. Researchers commonly apply this measure for topic identification [280], social opinion mining [281], and community detection [282]. We used it to calculate influence as it considers not just the number of connections a user has but also the importance of those connections. By assigning greater weight to users linked to other influential users, it captures how influence spreads in a network.

We calculate the centrality score for each subreddit independently before aggregating the results. This approach allows us to account for each subreddit's unique characteristics and interaction patterns. Given that different subreddits, such as *wallstreetbets*, *Stocks*, or *Dividends*, focus on distinct investment strategies and market discussions, users may have varying degrees of influence depending on the context. For instance, due to his speculative or sarcastic posts, a highly influential user within *wallstreetbets* may not carry the same weight in Dividends, which often promotes more severe and conservative discussions. By calculating the *CIS* within each subreddit separately, we capture the influence of a user concerning the specific context and dynamics of that community.

$$CIS(u,s) = C_{G_s}(u) \times log(M(s)) \tag{13.1}$$

Where G_s represents the subgraph derived from subreddit s, which includes only the submissions and comments related to s, along with the users who authored them. $C_{G_s}(u)$ represents the centrality score for the user u, in our case the PageRank.

The centrality value is then normalised by the number of members of the subreddit ($M(s)$); it helps to account for the scale differences between subreddits. By applying logarithmic normalization, we mitigate the disproportionate impact of subreddit size on centrality scores. This adjustment provides a more balanced comparison across subreddits of varying sizes and ensures that users from smaller subreddits with fewer members are no longer unfairly disadvantaged while still accounting for the fact that larger communities have more interaction. The scores for each subreddit are aggregated to provide a broader measure of a user's overall influence.

$$CIS(u) = log\left(\frac{k_u}{t}\right) \sum_{s \in S} CIS(u,s) \tag{13.2}$$

Where S is the set of the subreddit. In the equation 13.2, the normalization factor is represented by the logarithm of the ratio of a user's karma k_u to the time t, in terms of days they have been active on Reddit. The karma is a weight for calculating influence and measuring a user's reputation within the Reddit community. The variable t indicates the time the user actively posts on the platform, measured in units such as days or weeks. This normalization is essential as it helps to diminish the disproportionate influence of users with high karma who have been on the platform longer than newer users.

Sentiment Engagement Score (SES) This metric combines two fundamental aspects of user interactions on Reddit: the sentiment expressed in posts and the level of engagement they receive. Using these two elements, we aim to provide an understanding of how user contributions shape discussions. For each user u, we calculate the $SES(u)$ as:

$$SES(u) = \frac{\sum_{p \in P_u} sent_p \times ENG(p)}{\sum_{p \in P_u} ENG(p)} \tag{13.3}$$

where $sent_p$ represents the sentiment value for post p, which varies from -1 (negative value) to +1 (positive value) and is used to weight the engagement level of the specific post. $ENG(p)$ represents the engagement level of post p and it is defined as $\left(1+\frac{\text{upvotes}-\text{downvotes}}{\text{upvotes}+\text{downvotes}+1}\right)$. The literature proposes numerous engagement metrics for social network analysis [116, 283]. Specifically, we define engagement as a factor that adjusts the impact of the sentiment based on the level of community engagement. We calculate the numerator as the difference between upvotes and downvotes, and the denominator, which includes the total number of upvotes and downvotes plus one, normalizes the engagement and avoids division by zero for posts that have not a vote. The key idea behind this factor is that it amplifies or attenuates the sentiment based on how much support a post receives from the community. If a post has a high net engagement (more upvotes than downvotes), the sentiment is amplified, whether positive or negative. In contrast, if the post has low or negative engagement (more downvotes than upvotes), the sentiment is diminished, pushing its impact toward zero. This demonstrates that strong community support increases the relevance of the sentiment, while a lack of support or disagreement (downvotes) reduces its weight. The formulation of *SES* integrates both the sentiment expressed in a user's posts and the level of engagement those posts receive. This dual captures both the emotional tone and community reception of content.

Content-based Centrality score (CbC) Now we define the final formula for the influence score as follows:

$$CbC(u) = \alpha \times CIS + (1-\alpha) \times SES \tag{13.4}$$

The parameter α is a weighting factor that balances the importance of *CIS* and *SES* in calculating the final influence score. By adjusting α, it can prioritize either the structural influence of the user or their sentiment-based impact. This dual-layered approach ensures that this final score reflects localized impact within individual subreddits and the user's wider influence across the platform, offering a more accurate and context-sensitive evaluation.

The versatility of the *CbC* allows researchers to analyze a user's influence on the broader Reddit community and within the context of specific subreddits or individual stocks. One can isolate and evaluate the user's impact within a particular subreddit or stock discussion using the individual metrics defined in equations 13.1 and 13.3. This adaptability makes the *CbC* a valuable tool for understanding how user interactions and sentiments shape financial conversations on the platform.

13.3 Experiment

This section details the experimental analysis made on the Dataset, described in Section 13.3.2, following the experimental protocol designed in Section 13.3.

13.3.1 Experimental Protocol

The aim of our evaluation is twofold:

1. Comparing the *CIS* component with other well-established centrality metrics used for social network analysis to establish a benchmark for understanding how incorporating sentiment analysis can enhance the identification of relevant users.

2. Conduct a human assessment to evaluate how the proposed approach, which integrates post sentiment and engagement, enhances the identification of key users.

For our experiments, we use Reddit data to generate a graph on which centrality measures are calculated. Sentiment analysis is performed using a Roberta-based pre-trained model presented in [284, 285] and publicly available on the HuggingFace platform[3]. This model is a pre-trained transformer

[3] `https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest`

model specifically fine-tuned for sentiment analysis on Twitter data (∼124M tweets from January 2018 to December 2021). It is based on the RoBERTa (Robustly Optimized BERT Pretraining Approach) architecture and was trained using a large corpus of tweets. The model is designed to classify text into sentiment categories like positive, neutral, and negative, making it well-suited for tasks involving social media sentiment analysis [285]. All computations were carried out on a machine equipped with NVIDIA A10G Tensor Core GPUs[4].

13.3.2 Data acquisition and cleaning

To obtain the data, we utilized the PRAW[5] (Python Reddit API Wrapper) library, which provides a convenient interface for accessing Reddit's API. We leveraged the search function within PRAW to gather posts related to specific stocks by querying both their stock tickers (e.g., GME) and their full names (e.g., GameStop).

Selected stocks of interest. Given Reddit's prominent role in shaping retail investor sentiment and its potential influence on market behavior, our study hones in on specific stocks. We selected 50 stocks to represent various sectors and market phenomena comprehensively. Many of these stocks, such as GameStop (GME) and AMC, have garnered significant attention as *meme stocks*, illustrating the power of social media in influencing retail investor behavior. These stocks are often characterized by heightened volatility and rapid price movements, making them particularly interesting for studying sentiment and user influence in online communities. Additionally, we included stocks from emerging industries, such as electric vehicles (e.g., BYD) and biotechnology, like Novavax (NVAX) and Moderna (MRNA). Furthermore, we considered stocks in the tech sector, such as Advanced Micro Devices (AMD) and the newly popular Open (OPEN), which are at the forefront of technological innovation. Some stocks, like General Motors (GM) and Starbucks (SBUX), represent established companies with robust market positions, contrasting the more speculative meme stocks. Additionally, stocks like Clover Health (CLOV) are notable for their appeal to younger, more speculative investors looking for high-growth opportunities, while stocks like Silver (SLV) appeal to those seeking safe-haven investments during periods of economic uncertainty. By encompassing a wide range of sectors and investment types, we aim to provide a comprehensive view of the discussions taking place in online financial communities and their potential impact on market movements. By executing search queries for each stock ticker and its corresponding full name (e.g. GME and GameStop), we retrieve posts that mention these stocks within their content. This dual-query approach ensured comprehensive coverage of discussions related to the stocks of interest, as users may reference the stocks either by

Table 13.1: Information about different subreddits including the number of downloaded submissions and posts

Subreddit	Creation on	Members	Submission	Post
wallstreetbets	31/01/2012	16,735,396	11,640	1,198,535
Stocks	27/06/2008	7,781,666	8,906	256,212
economics	25/01/2008	4,669,690	1,373	12,163
StockMarket	09/07/2008	3,064,063	8,556	88,280
investing	15/03/2008	2,756,619	6,165	138,733
finance	13/03/2008	1,985,153	739	6,848
pennystocks	31/12/2008	1,923,312	7,573	149,720
Dividends	30/01/2009	592,104	2,291	44,331

their ticker symbols or by their full names.

Selected subreddit of interest. In our data acquisition process, we limited our focus to comments that were direct responses to submissions and replies to those comments instead of delving deeply into the different levels of the discussion. We made this decision to maintain a clear and focused dataset that directly reflects user interactions with the original posts. By concentrating on first-level replies, the data collected provides straightforward insights into how users respond to specific stock-related content without the added complexity of nested discussions.

Tab. 13.1 provides an insightful overview of the chosen subreddits, showcasing their creation dates, membership sizes, and submissions and posts from Reddit during the reporting period. Notably, *wallstreetbets* stands out with an overwhelming 16+ million members, making it the largest and most influential subreddit in this selection. It also boasts over 11640 submissions and nearly 1.2 million posts, indicating high engagement and activity. Other significant subreddits include *Stocks* and *economics*, with 7.7 million and 4.6 million members, respectively, exhibiting substantial interaction among users. In contrast, smaller subreddits like *Dividends* and *finance* have more niche communities with lower activity levels, reflected in fewer submissions and posts. This diverse range of subreddit sizes and engagement levels suggests that some, like *wallstreetbets*, dominate the discourse and attract widespread participation, others serve more specialized financial interests, making them valuable in understanding specific market segments.

We focused our analysis on the period from 2015 to 2024, considering only users who registered on Reddit starting in 2015. This time frame ensures that our data reflects the evolving nature of financial discussions and market dynamics over recent years, particularly in the context of significant events that have influenced investor behavior, such as the rise of meme stocks and increased retail investor participation. To analyze the impact of users on the market effectively, we filtered for the most active participants, specifically selecting those with a minimum of 100 comments during the observation period. While this threshold is empirical, it is well-suited for our purpose, as it allows us to focus on users who have demonstrated a sustained level of engagement in discussions. It increases the likelihood that their contributions hold relevance and influence within the community. Consequently, we retained only the comments made by these users.

After collecting and cleaning the data, we constructed a graph following the schema outlined in Section 13.2.1. Figure 13.3, we present an extract of the complete graph.

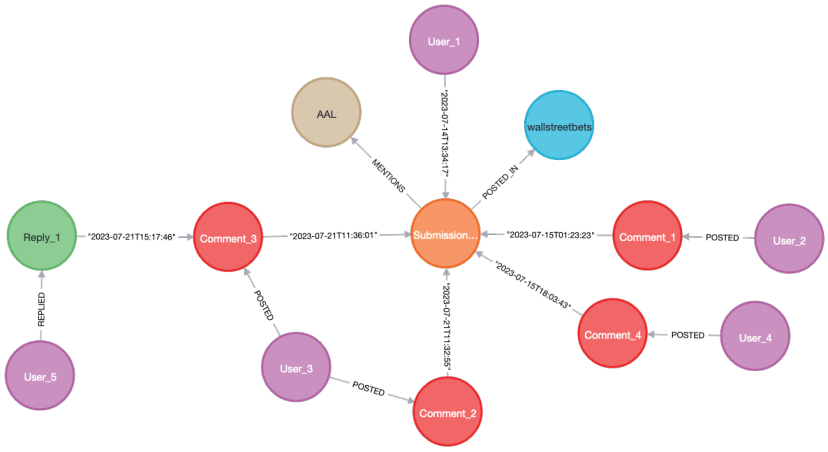


Figure 13.3: Extraction of the graph with entities involved and their connections.

[4] https://www.nvidia.com/en-us/data-center/products/a10-gpu/
[5] https://praw.readthedocs.io/en/stable/

Post Sentiment Classification. To assign a sentiment score to each post, we create a pipeline to associate each text with a score. After a light preprocessing phase, where we removed URLs and any HTML codes but kept emojis (as the model can handle them), we fed the preprocessed texts into the pre-trained sentiment model used. As output, we get the probabilities of the three sentiment classes: positive (POS), neutral (NEU), and negative (NEG). To obtain a single sentiment score ranging from -1 to +1, we applied a simple transformation by calculating the difference between the probabilities of the positive and negative classes. This approach yields a continuous sentiment scale, where values closer to -1 indicate strong negativity, values near +1 suggest strong positivity, and values around 0 represent neutrality. This method provides a view of sentiment rather than relying solely on discrete class predictions, making it especially useful in the context of user interactions and engagement within the community.

13.4 Results

As introduced in Section 13.3.1, the first step involved computing the *CIS* for each user, as detailed in Equation 13.2, and conducting a comparison with other widely recognized centrality measures found in the literature. Specifically, for the *CIS*, we use for the normalization parameter t the number of days from the user's first post to the last day of our data collection (for a total of 3,456 observation days). It ensures that the influence of both newer and long-standing users is fairly represented, adjusting for the time they have been active on the platform.

The traditional metrics used for comparison in our analysis included degree centrality, closeness centrality, and PageRank, which widely used in the literature for measuring users' influence in social networks [124, 123, 116]. In Reddit discussions, degree centrality identifies active users who are more visible in the community because they create more submissions and interact more often with other submissions. On the other hand, closeness centrality reflects the overall proximity of the user within the network. It calculates the average shortest path from a given user to all other users, which is a proxy for how quickly this user can interact with others.

PageRank consider both the quantity and the quality of connections. In the Reddit network, where discussions exhibit a thread-based (hierarchical) structure, PageRank delivers more robust results because the calculation is performed with a simulation of the likelihood of a random walk, allowing for traversing the network.

Figure 13.4 displays the Kendall Tau correlation coefficient [286] between the various metrics. Notably, traditional centrality measures exhibit a strong correlation in their values since they assess the complete graph. In contrast, our *CIS* metric focuses on the subgraphs of individual subreddits and aggregates the findings, enabling it to distinguish scenarios based on specific subreddits. This method enhances the sensitivity and relevance of our metric in identifying user influence within particular contexts. We selected the Kendall test due to its effectiveness in evaluating correlations among ordinal variables and its capacity to manage tied values, making it especially appropriate for our analysis, where centrality measures may yield similar or identical results. The p-values associated with the various correlations are nearly zero, indicating a statistically significant relationship between the metrics.

The distributions of all four centrality metrics are illustrated in Figure 13.5, which reveals a strong skew toward lower values across all metrics. It indicates that most users exhibit low influence or limited connections within the Reddit network. The *CIS* metric, which captures local influence within subreddits, shows that only a small fraction of users emerge as highly influential. Similarly, the *Degree Centrality* and *PageRank* confirm that a few individuals hold the majority of influence. In contrast, *Closeness Centrality* highlights a more pronounced separation between central and peripheral users in the graph.

Finally, for each user, we calculate the *CbC* across the entire observation period using the equa-

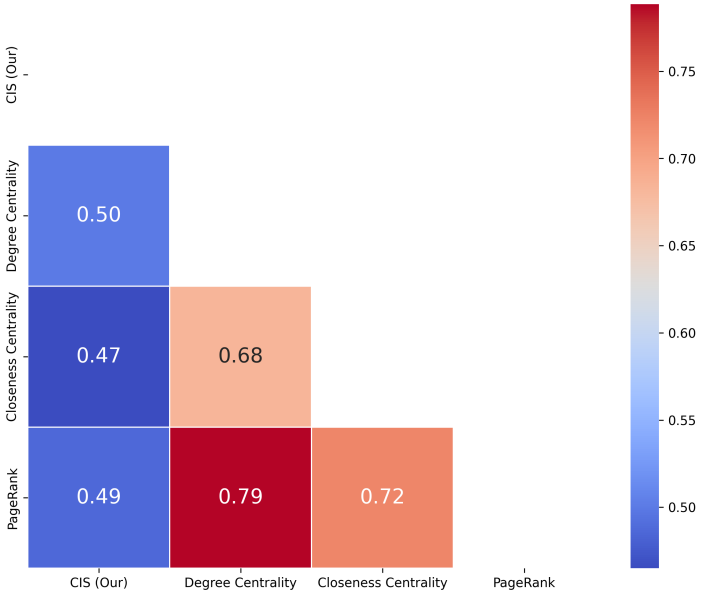


Figure 13.4: Kendall correlation heatmap between different centrality metrics

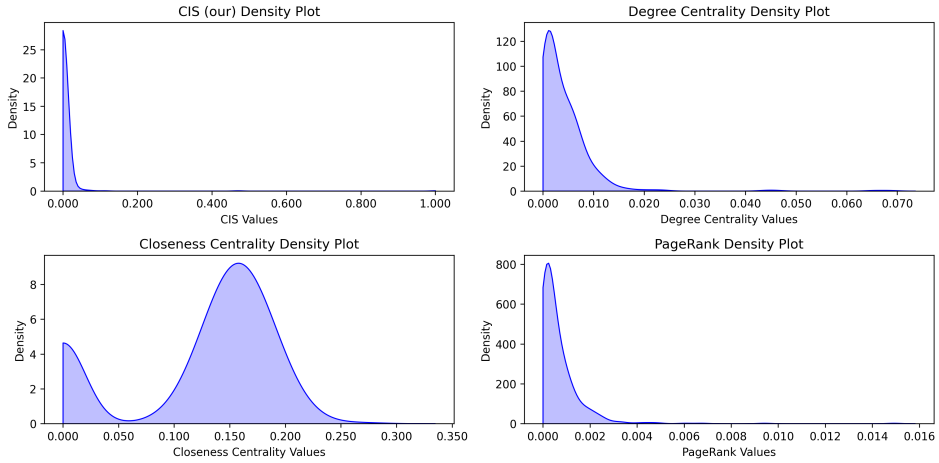


Figure 13.5: Density plots of the four centrality metrics

tions 13.2 and 13.3. The parameter α is set to 0.5, giving equal weight to both components of the score so that the user influences the subreddit community, and their impact on stock discussions is considered equally in the final score. This balanced weighting ensures that neither the community interactions nor the sentiment towards stocks disproportionately influences the overall score.

13.4.1 Human evaluation

Given the absence of a baseline and the complexity of measuring user influence on Reddit through algorithmic metrics, we opted for a human evaluation to validate our results rather than relying solely on automated methods. While computational metrics are effective at quantifying influence, they may overlook important contextual nuances—such as whether a post is genuinely relevant to financial markets or simply general discussion. This kind of semantic understanding is challenging to capture automatically. A human evaluation allows us to better account for these subtleties, providing a qualitative assessment to complement the quantitative scores.

We applied human evaluation to assess two key aspects: (1) whether individual posts had any

real connection to the financial world, and (2) how interesting and trustworthy evaluators perceived those posts. To do this, we created groups based on the rankings from our four influence metrics. Specifically, we selected the top 10 users for each of the metrics and created 10 distinct groups, where each group contained the users ranked in the same position across the four metrics. For example, Group 1 included the users ranked first in all four metrics, Group 2 included those ranked second, and so on.

For each user in a group, we selected one of their posts and asked the evaluators the first question: *(Q1) Is the post related to financial markets, or is it merely general conversation? (Yes/No)* This step was essential to ensure that we evaluated user influence in a contextually meaningful way, particularly for our analysis of financial discussions.

After reading all four posts in a group, evaluators were then asked the second question: *(Q2) Rank the posts according to the interest and trustworthiness you found in its content regarding the stock market. (rank 1 - very interesting/reliable, rank 4 - not very interesting/reliable).* This process was repeated for all ten groups, ensuring that each of the top 10 users across the different metrics was assessed. This method allowed us to gather human judgment on the relevance and quality of posts for different influential users, offering a comparative perspective on the four metrics. In this way, we gather human judgment on the relevance and quality of posts for influential users, offering a comparative perspective on the four metrics. By comparing users who held the same rank across different metrics, we aimed to see how the user's score assigned by the metrics reflected the real interest in the user's content. This process was repeated across all 10 groups, giving us a detailed comparison between users ranked similarly by the different metrics, and providing valuable insights into the nuances of user influence on Reddit's financial discussions.

The first question helps us understand whether the suggestion of the most relevant users is based on their contribution to a financial discussion. In this case, 100% of the posts ranked by *Closeness Centrality* were relevant to the financial sector, while for *PageRank* and *CbC*, this percentage was 90%, and for *Degree Centrality*, it dropped to 70%. These results suggest that *Degree Centrality*, which only considers direct connections, may be misleading when identifying key financial-related content.

Table 13.2: Metrics human evaluation

Metrics	top@1	top@3	top@5	top@10
Degree centrality	0	0	1	1
Closeness centrality	**1**	1	1	2
PageRank	0	0	0	1
CbC (our)	0	**2**	**4**	**6**

Table 13.2 shows how each metric performs in terms of suggesting relevant users. Each *top@k* column represents the number of times, considering the top *k* users, that a post ranked by that metric was deemed the most relevant. For example, in the *top@3* column, the *CbC* metric reaches a value of 2, meaning that the content from the top 3 users was rated as the most relevant in 2 out of 3 instances. The results show that our *CbC* metric identified more pertinent users of the top@3, top@5, and top@10 groups than the other centrality metrics. Specifically, *CbC* outperformed the different measures, which remained more limited. Obtained results confirm the effectiveness of our metric in identifying potentially influential users compared to traditional methods like *Degree Centrality* and *PageRank*, which achieved lower results.

13.5 Conclusion

The *Content-based Centrality score* (CbC) provides a comprehensive measure of a user's influence on Reddit, aggregating different metrics to reflect their impact on both specific subreddits and individual stocks. We calculate this score by combining the *Centrality-based Influence Score* (CIS) and the *Sentiment Engagement Score* (SES), which together assess the user's structural position within the subreddit network and their sentiment-based engagement with stock discussions. In our analysis, we compared the *CbC* with traditional metrics such as *Degree Centrality*, *Closeness Centrality*, and *PageRank*. While the latter metrics provided valuable insights, they often failed to capture the nuanced influence of users in financial discussions. For instance, *Degree Centrality* showed a lower relevance percentage in financial posts (70%). Moreover, human evaluation confirmed the effectiveness of the *CbC* in identifying influential users whose contributions align with significant stock discussions. These results underline the importance of integrating structural and sentiment-based metrics, enhancing our understanding of influence dynamics within the Reddit community. The *CbC* offers researchers the flexibility to assess a user's influence across the entire Reddit community and within specific subreddits or for desired stocks. By leveraging the *CIS* and *SES* metrics, researchers can focus on a user's impact in a particular contest. This adaptability positions the *CbC* as an essential tool for analyzing how user interactions and sentiments influence financial discussions on the platform. Ultimately, the *CbC* not only serves as a robust analytical tool but also provides actionable insights for investors and researchers looking to navigate the complexities of financial discourse on social media platforms.

Chapter 14

Learning Across Modalities: A Systematic Survey of Multimodal Models for Financial Analysis

Financial markets are complex socio-technical systems in which heterogeneous sources of information coexist and interact. Beyond numerical price signals, market dynamics are increasingly influenced by textual information from news and social media, as well as by relational structures linking assets, companies, and investors. While recent advances in natural language processing have shown that vector space models can effectively extract sentiment and trends from unstructured text, these approaches remain inherently limited by their unimodal perspective and are unable to capture the full complexity of financial phenomena

This observation motivates a growing shift toward multimodal learning, which aims to jointly model multiple data modalities, such as market time series, textual data, and graphs—in a unified predictive framework. Rather than treating each information source independently, multimodal approaches explicitly account for cross-modal interactions, temporal dependencies, and complementary signals, offering a more faithful representation of real-world financial systems. As a result, multimodal learning has recently emerged as a powerful paradigm for financial forecasting and decision support.

In this context, we present a comprehensive survey of multimodal learning methods for financial forecasting[1] presents a unified taxonomy for multimodal financial forecasting models, structured along four key dimensions: input modalities, modeling architectures, fusion strategies, and predictive tasks, as shown in Fig. 14.1. Using this taxonomy, we conduct a systematic review of 35 representative works published between 2018 and 2025, highlighting methodological trends, design choices, and performance patterns. Our analysis identifies persistent challenges, including temporal misalignment, modality imbalance, missing or noisy data, and limited cross-market generalization. We also discuss emerging trends and promising research directions, such as adaptive fusion, incomplete modality learning, and the integration of large language models and temporal graph neural networks, aiming to bridge methodological innovation with domain-specific requirements. This work is supported by the Italian Ministry of University and Research (MUR) within the PRIN2022—ISALDI: Interpretable Stock Analysis Leveraging Deep multimodal models (CUP: E53D23008150006).

[1] Under final review on Information Fusion Journal

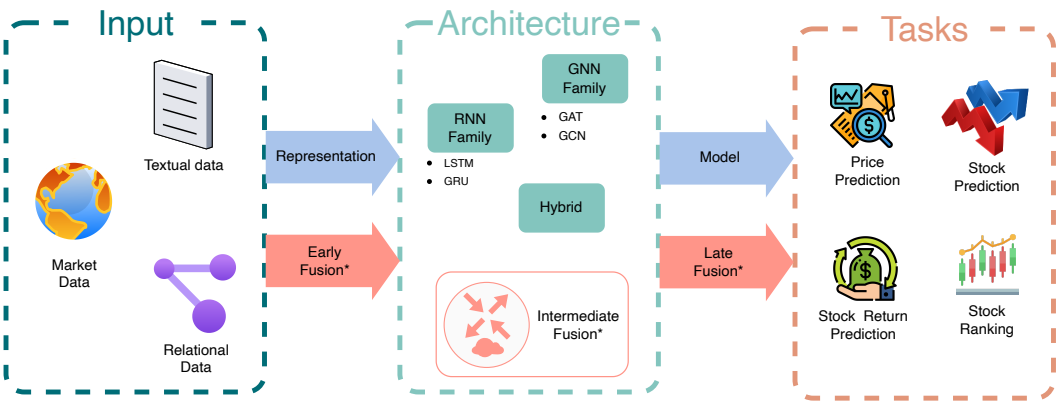


Figure 14.1: General flow of the framework: Different input modalities are processed to produce models capable of performing various tasks. Elements marked with * indicate optional components.

14.1 Introduction

Financial forecasting has long been a central focus in academic research and industry, predicting market trends, asset returns, and risk dynamics to inform trading strategies and investment decisions. Traditionally, forecasting models have relied predominantly on historical market data, such as time series of price and volume, to capture patterns and generate future predictions [73, 74]. However, financial markets are influenced by various exogenous factors, including macroeconomic signals, geopolitical events, investor sentiment, and corporate disclosures, which cannot be fully captured by quantitative indicators alone [75, 76].

The growing availability of heterogeneous financial data sources, encompassing both structured inputs (e.g., financial statements, technical indicators) and unstructured content (e.g., news articles, regulatory filings, social media posts), has fueled interest in multimodal approaches. This shift has spurred interest in multimodal approaches, which aim to integrate heterogeneous data streams to better reflect the complex nature of financial systems. Recent studies confirm the growing effectiveness of multimodal fusion in stock prediction across sectors [77, 78, 79]. Advanced approaches also incorporate causal reasoning and saliency-aware augmentation to improve prediction robustness and exploit macroeconomic signals [80, 81]. By combining multiple modalities, such models can uncover complementary signals, capture latent dependencies, and enhance robustness to market shocks and behavioral dynamics.

Recent literature emphasizes the importance of integrating multiple sources of information to improve predictive performance and model robustness [86]. Additionally, hierarchical fusion strategies and large-scale neural network-based frameworks have been shown to enhance both predictive accuracy and interpretability [87, 88]. These sources include not only structured time series but also textual news, graph-based relationships, and alternative data streams.

The rise of deep learning has further accelerated this trend, enabling the development of architectures that are capable of extracting modality-specific features and combining them in flexible and scalable ways. In this context, financial forecasting has become an increasingly multimodal task, involving the interplay of temporal models (e.g., LSTMs [84] for time series), language models (e.g., BERT [3] for text), relational architectures (e.g., GNNs [85] for asset graphs), and fusion mechanisms that align these representations into a unified decision-making framework. Recent works highlight the systematic review of multimodal fusion techniques and knowledge graph reasoning as emerging foundations for financial forecasting [287, 288, 289, 290]

Despite the growing body of work in this area, the literature remains fragmented in terms of archi-

tectural choices, data modalities, fusion strategies, and evaluation settings [89, 90, 91]. This survey aims to provide a comprehensive overview of recent advances in multimodal financial forecasting, organizing contributions along a set of analytical dimensions, and highlighting key trends, challenges, and future directions.

14.1.1 Motivation and Contribution

Multimodal learning has recently gained prominence in financial forecasting, enabling models to integrate diverse information sources such as time-series data, textual news, social media sentiment, and fundamental indicators. While each modality can independently offer predictive value, their combination yields a richer representation of market dynamics. Empirical studies confirm that multimodal models often outperform unimodal counterparts in tasks such as stock movement prediction, risk assessment, and volatility estimation [291, 292, 78].

Despite this growing body of work, the field remains fragmented. Studies differ significantly in terms of input modalities, fusion mechanisms, modeling techniques, and evaluation protocols. Existing surveys have made valuable yet focused contributions: Wang et al. [90] provide a thorough categorization of *graph neural network methods* and their financial applications, while Patel et al. [89] offer a systematic review of *graph-based prediction models* specifically for stock market forecasting, highlighting results and open issues.

However, these works primarily focus on a single architectural paradigm (GNNs) and do not provide a systemic multimodal perspective. A comprehensive and unified taxonomy encompassing the full spectrum of data modalities (market, text, relations), model architectures (RNNs, GNNs, Transformers, hybrids), fusion strategies (early, intermediate, late), and predictive tasks is still missing. This gap makes it difficult to compare approaches across architectural families, uncover shared design principles, and identify cross-cutting research challenges.

To address these gaps and the increasing interest in using large language models (LLMs), graph-based architectures, and temporal reasoning across multiple modalities [89, 287], this survey provides more than a structured overview; it offers a novel taxonomic framework to map the landscape, consolidate insights, and provide a structured reference for researchers and practitioners. This work addresses that need by providing a structured overview of state-of-the-art multimodal forecasting models, analyzing their methodological foundations, highlighting recurring challenges, and identifying emerging research trends.

Contributions. To address the lack of a unified perspective in the growing literature on multimodal financial forecasting, this survey offers a structured and in-depth analysis of recent contributions to this field. In particular, we aim to clarify how different works approach the integration of heterogeneous data sources, which architectural solutions are most common, and what challenges remain open in this rapidly evolving field. Building on this motivation, we summarized our main contributions as follows:

1. **Taxonomy-Based Analysis**: A novel taxonomy that classifies multimodal forecasting models across four dimensions: input modalities, modeling architectures, fusion strategies, and predictive tasks. This taxonomy, presented in Section 14.3, facilitates a consistent comparison and reveals common patterns and design trade-offs.

2. **Systematic Literature Review**: A comprehensive review of 35 papers published between 2018 and 2025, each analyzed through the lens of the proposed taxonomy. The review spans various data combinations, learning architectures, and forecasting objectives.

3. **Insights and Research Directions**: A synthesis of key technical challenges—such as temporal misalignment, missing or noisy data, and modality imbalance—that are commonly addressed in the literature. We discuss these challenges in detail in Section 14.4 and outline potential directions for future research.

14.1.2 Scope

We conducted a structured literature search on Google Scholar[2] focusing on recent advances in multimodal and graph-based models for stock market prediction. The search combined domain-specific keywords and Boolean operators to capture a broad set of relevant works addressing multimodal representation and relational modeling.

Keyword queries included: *"multimodal model for stock forecasting"*, *"multi-modality graph model for financial forecasting"*, and *"fusion input for financial forecasting model"*. Additionally, compound queries such as *["graph neural networks" OR "knowledge graphs" OR "relational modeling"] AND ["stock market prediction" OR "stock market forecasting" OR "stock movement prediction"]* expanded the search scope. This approach allowed us to retrieve papers employing multi-modalities-based architectures for financial prediction, even when different terminologies were used.

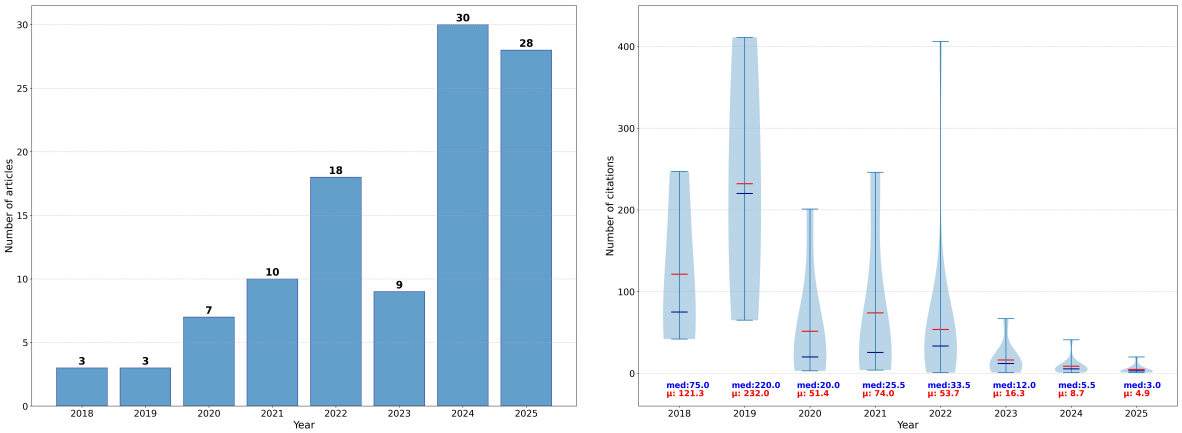
Our search yielded 108 papers published between 2018 and 2025, comprising journal articles, conference papers, and influential arXiv preprints. Fig. 14.2a displays the temporal distribution of publications, revealing a marked increase in recent years despite a slight dip in 2023. Fig. 14.2b further complements this analysis by showing the citation distribution per year.

Selected journal papers are mostly from Q1-ranked venues, with a few Q2 exceptions. Conference papers come from venues ranked A. We included some non-peer-reviewed preprints due to their relevance for our study.

The chosen time frame reflects the rise of enabling technologies such as graph neural networks (2016-2017) and transformer-based language models like BERT (2017), which have catalyzed multimodal approaches in financial forecasting.

The combination of these architectures with structured market data and alternative modalities (e.g., news, social signals) laid the foundation for the development of multimodal approaches. The chosen time window thus captures both the emergence of enabling technologies and the subsequent evolution of multimodal methods for financial forecasting.

Papers were included based on two criteria: (i) use of multimodal models combining at least two types of input data beyond the target variable itself (e.g., market prices plus textual or event data); and (ii) the application to stock prediction tasks, including price forecasting, movement classification, or return estimation. After screening, 35 papers met these criteria and form the basis of our review.



(a) Distribution of articles over the years.

(b) Distribution of citations over the years.

Figure 14.2: Analysis of the distribution of papers by year and citation count.

[2] https://scholar.google.com/

14.2 Fundamental Concepts

14.2.1 General Multi-Modality Forecast Prediction Framework

Multimodal models are systems designed to process and integrate data from multiple heterogeneous sources or modalities, such as text, time series, images, or graphs. In the context of financial forecasting, they aim to leverage diverse information—such as historical stock prices, financial news, social media sentiment, and inter-company relationships—to enhance predictive accuracy and model robustness.

In recent years, multimodal learning has emerged as a promising paradigm for stock market prediction, enabling the integration of heterogeneous data sources that capture different dimensions of financial markets [293, 294]. While traditional models rely primarily on historical price data, multimodal approaches enrich the learning process by incorporating textual, relational, and structured signals—leading to improved robustness and predictive performance, particularly under volatile or complex market conditions.

Underlying these models is the core hypothesis that different modalities provide complementary perspectives [295, 296]. For instance, while numerical market data captures quantitative price movements, textual sources (e.g., news articles or social media) offer insights into sentiment and external events that may not yet be reflected in market prices. In parallel, graph-based structures can model static or dynamic relationships between entities—such as industry dependencies or supply chain interactions—adding a valuable relational dimension. By combining these perspectives, multimodal models are able to generate more informed and context-aware predictions.

Several recent studies exemplify the benefits of such integrative strategies. Sawhney et al.[297], in their FAST model, combine tweets and financial news with historical price data using transformer-based architectures and attention mechanisms. Similarly, Qian et al.[298] propose MDGNN, a multi-relational dynamic graph neural network that integrates temporal price patterns, static relations, and evolving graph structures. Zhao et al. [299] employ a hybrid architecture that fuses textual and numerical signals to build a robust representation of market dynamics.

However, integrating such diverse modalities is non-trivial. It often requires architectural innovations for encoding, aligning, and fusing multimodal signals effectively. These challenges have made multimodal modeling a central and rapidly evolving theme in financial AI research.

Figure 14.3 illustrates the general flow underlying multimodal models for financial prediction, structured into four key components: **Input**, **Architecture**, **Fusion**, and **Tasks**. These components represent the fundamental stages involved in designing and evaluating multimodal systems.

- **Input:** Multimodal models typically process heterogeneous data sources, including market data (e.g., stock prices), textual data (e.g., financial news or social media), and relational or structured data (e.g., graphs of company relationships or supply chain relationships). This diversity of inputs reflects the real-world complexity of financial markets and is essential for capturing complementary signals.

- **Architecture:** Once the data are represented in a usable format, different neural architectures are applied. These may include feedforward neural networks (FNN), recurrent neural networks (RNN) or graph neural networks (GNNs). Hybrid approaches that combine multiple architecture families are also increasingly adopted.

- **Fusion:** A critical design choice in multimodal modeling is how and when to integrate different modalities. As the figure shows, fusion strategies can be applied at various stages–early, intermediate, or late–each with different implications for model capacity and interpretability.

- **Tasks:** The final goal of these models is to perform downstream financial prediction tasks, such as price prediction, stock movement classification, pattern recognition, or stock ranking. These

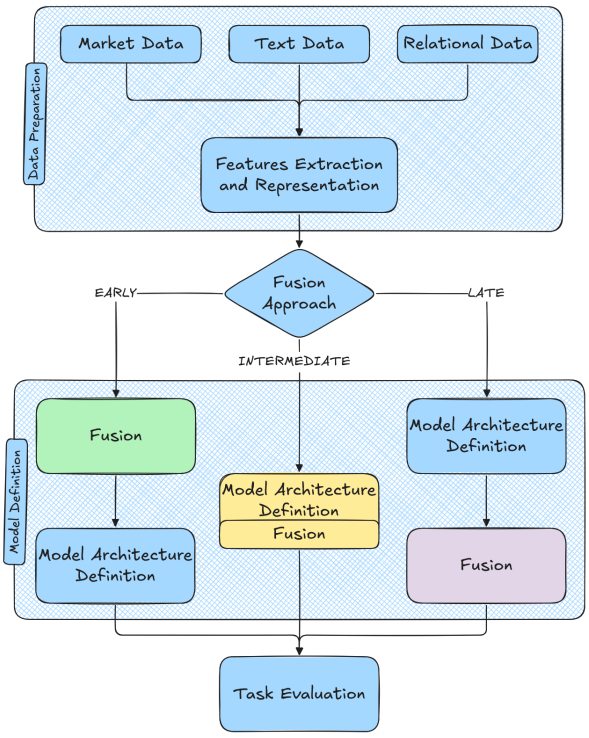


Figure 14.3: Workflow of a multimodal financial forecasting pipeline, showing data preprocessing, exclusive selection of fusion approach (early, hybrid, late), model training, and evaluation.

tasks guide the evaluation and benchmarking of the multimodal models.

This conceptual framework guides the structure of our survey: in this section, we introduce each of these four components in detail, providing the necessary technical foundations. Subsequently, in Section 14.3, we use this framework to systematically compare and categorise analysed papers.

14.2.2 Data Modalities in Financial Forecasting

Multimodal financial forecasting models leverage heterogeneous data sources to better capture the multifaceted nature of financial markets. These modalities provide complementary information that, when properly integrated, can significantly enhance prediction accuracy and robustness.This section introduces two further main categories of data used, in addition to the textual data already discussed in previous sections.

Market Data

Market data consists of time-series signals that reflect, either directly or indirectly, asset trading activity. The underlying rationale for using market data is that asset prices often exhibit recurring patterns and cyclical behaviors that machine learning models can learn to exploit. Market Data can be broadly categorized into two main types: raw trading data and technical indicators. Raw trading data typically includes Open, High, Low, and Close (OHLC) prices along with traded volumes—fundamental inputs available for virtually every publicly traded asset. Figure 14.4 shows a typical representation of those values.

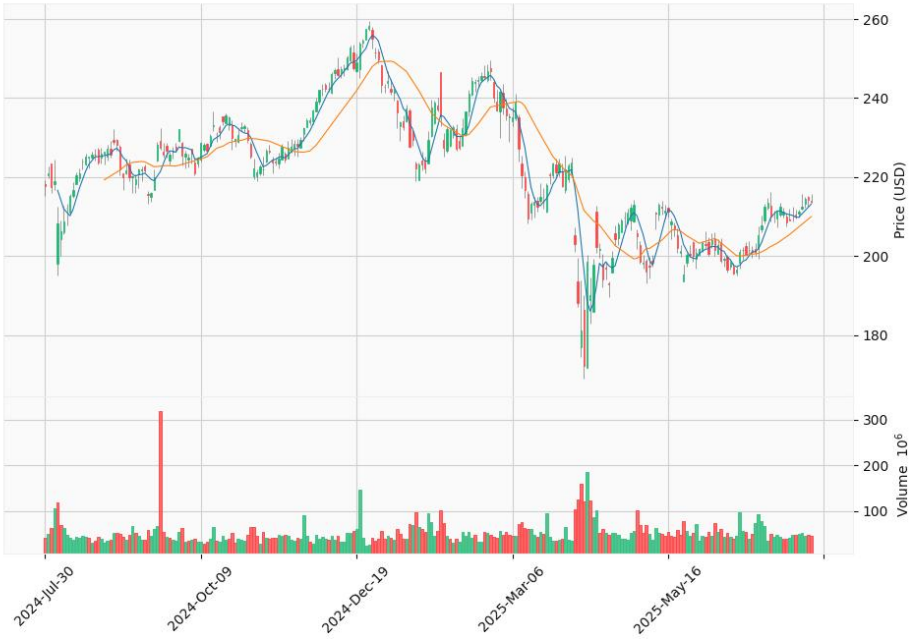


Figure 14.4: Candlestick chart for Apple Inc. (AAPL), displaying daily OHLC prices over the past 12 months. Green and red candles respectively indicate upward and downward movements in closing prices. The chart includes 5-day and 20-day moving averages to highlight short- and medium-term trends. At the bottom, volume bars show daily trading volume, offering insight into investor activity and market pressure.

Technical indicators, on the other hand, are mathematical transformations or summaries derived from OHLC data. Common examples include Moving Averages (SMA, EMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Bollinger Bands, and stochastic oscillators [89]. These indicators are designed to capture patterns related to trend strength, momentum, volatility, and mean reversion.

In high-frequency trading or intraday forecasting settings, more granular information is often incorporated, such as bid-ask spreads, order book dynamics, or tick-level price updates. While time granularity can range from milliseconds to daily intervals, daily frequency remains the most common in academic literature due to data availability and computational constraints.

From a modeling perspective, traditional statistical methods like ARIMA [73] and GARCH [300] have been widely used on univariate price series. However, deep learning approaches—such as LSTM networks or Transformer-based architectures—are better suited to capturing complex, multivariate, and nonlinear temporal dependencies [301].

How to represent market data. To be incorporated in multimodal architectures, market data can be represented in various ways depending on the desired level of abstraction and the predictive task. A simple approach involves concatenating raw OHLC values into a single input vector. For example, a fixed-length window of historical data can be flattened into a feature vector that preserves the temporal structure and is suitable for models like MLPs, LSTMs, or Transformers [302].

A more structured representation involves computing statistical or technical features over predefined intervals. These features—such as moving averages, volatility bands, or momentum scores—not

only reduce dimensionality but also offer domain-specific interpretability, making them popular in both traditional finance and deep learning settings [89, 90].

An alternative is to transform time series into graph structures, capturing their intrinsic geometry and temporal relationships. One notable method is the Visibility Graph Algorithm, which is widely employed (e.g [303, 304]) for financial time series analysis because it is not influenced by any algorithmic parameters and maps time series into scale-free graphs [305]. It converts a time series into a graph by treating each time point as a node and connecting pairs of points based on a visibility criterion, as shown in Figure 14.5: two nodes share an edge if no intermediate point obstructs the straight line between them. This transformation allows for the extraction of topological features—such as degree distribution or clustering coefficients—which can then be fed into graph-based or hybrid models.

Upon the definition given by Lacasa et al. [305], we can establish the following visibility criteria: two arbitrary data values (t_a, y_a) and (t_b, y_b) will have visibility, and consequently will become two connected nodes of the associated graph, if any other data (t_c, y_c) placed between them fulfills:

$$y_c < y_b + (y_a - y_b)\frac{t_b - t_c}{t_b - t_a}. \tag{14.1}$$

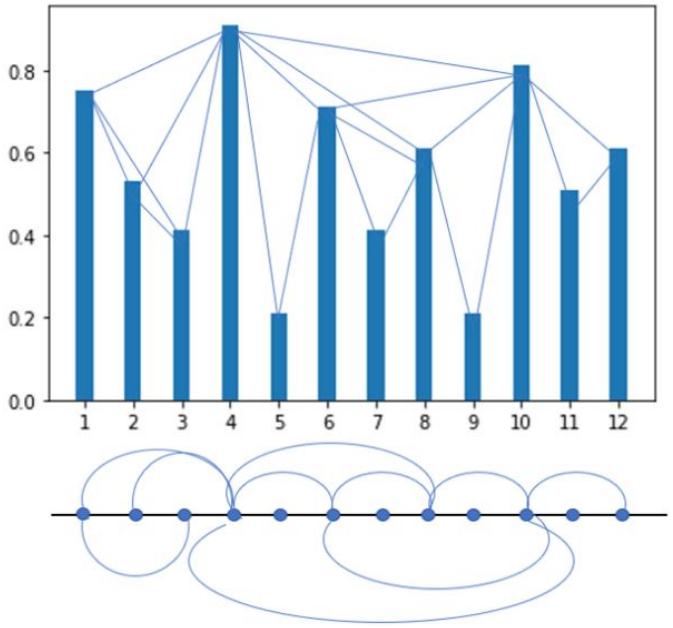


Figure 14.5: An example of the application of the visibility algorithm from [305].

These representations—ranging from raw sequences to engineered features and graph-based encodings—can be seamlessly integrated with other modalities (e.g., textual news, images, or social signals) to capture the multifaceted nature of financial markets in multimodal forecasting tasks.

Relational data

Relational data are essential in stock prediction models, as financial markets function as highly interconnected systems in which the performance of a single asset is often shaped by its links to other entities. Unlike market data or textual sentiment, relational data capture structural dependencies and latent correlations between companies, industries, and broader economic ecosystems.

One of the most common forms of relational information is the **industry or sector classification** of a company. Knowing which sector a company belongs to allows models to contextualize its performance, since companies within the same sector often react similarly to macroeconomic events or

policy changes, and strong correlations between their stock movements are frequently observed. For example, a decline in the energy sector caused by falling oil prices may simultaneously impact the stocks of multiple energy-related companies, regardless of their fundamentals.

Another important subset of relational data involves **corporate governance and ownership structures**. Information about board members, executives, or institutional investors—especially when individuals hold positions in multiple companies—can reveal non-obvious connections that influence market behavior. For instance, the appointment of a high-profile CEO to a new company may raise investor expectations based on their previous track record. Similarly, the presence of shared directors or cross-ownership links can induce indirect dependencies between firms that are not evident from price data alone.

Supply chain relationships and strategic partnership also constitute critical relational information. Disruptions in the supply chain of a key supplier can cascade and affect the financial performance of dependent customers. Modeling these networks of dependencies enables the capture of shock propagation across the economy. Geographic links, such as regional exposure or shared operational locations, also provide valuable insights, particularly during localized crises or geopolitical events.

How to Represent Relational Data. By their nature, these complex relationships are best modeled as graphs, where nodes represent entities such as companies, sectors, or locations, and edges capture various types of relations, including ownership, collaboration, or similarity. From these comprehensive graphs, specific components of interest are often extracted to focus analysis. For example, the correlation network between stocks highlights pairs or groups of assets with strong price co-movements, which can inform portfolio diversification or risk assessment. Other subgraphs might represent supply chain clusters or governance communities, enabling targeted analysis of propagation effects or influence patterns.

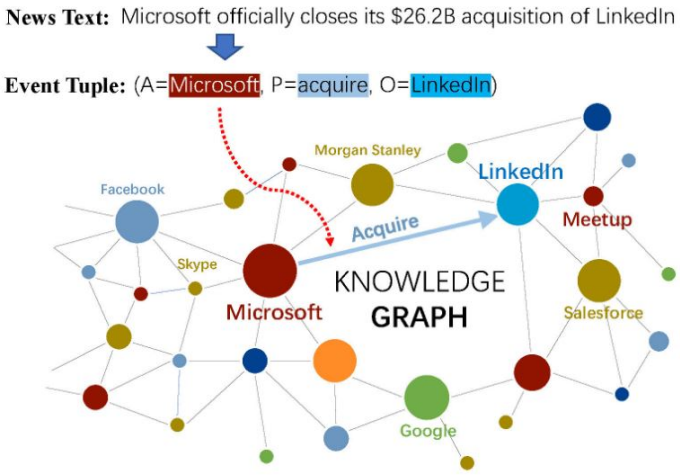


Figure 14.6: Example from [306] illustrates the analysis of associated events and how they affect relationships between stocks.

Events extracted from textual sources constitute another valuable form of relational data, as illustrated in Figure 14.6. Leveraging various event detection methods proposed in the literature [307], such information can be incorporated either as static features or as temporal profiles (e.g., updated quarterly). These features are frequently used as node attributes or edge weights within financial knowledge graphs and graph neural networks (GNNs), enabling more sophisticated relational reasoning [86, 90]

Overall, graph-based modeling of relational data provides a powerful framework to incorporate structural dependencies into financial prediction models, complementing traditional time series and textual data with rich, interpretable relational context.

14.2.3 Core Neural Architectures for Financial Data Processing

In multimodal financial forecasting, different neural network architectures serve as fundamental building blocks, each specialized in handling specific data types. These models—ranging from recurrent and convolutional networks to attention-based and graph-based architectures—are often individually applied to one modality (e.g., time series, textual information, relational data), and subsequently integrated into broader multimodal frameworks. Their modular design facilitates the extraction of modality-specific representations, which are then fused to capture cross-modal interactions and improve forecasting performance. This section presents the most widely used neural architectures in the financial forecasting literature, discussing their mechanisms, strengths, and typical applications.

Recurrent Architectures: LSTM and GRU

Recurrent Neural Networks (RNNs) have been widely adopted for modeling sequential data, particularly in the financial domain where time-dependent structures are critical. However, traditional RNNs often struggle to capture long-range dependencies due to the vanishing gradient problem. To overcome this limitation, more advanced architectures such as Long Short-Term Memory (LSTM) networks [84] and Gated Recurrent Units (GRU) [308] were introduced. These models have become fundamental components in financial forecasting tasks, including the prediction of stock prices, returns, and volatility [309, 310].

LSTM networks incorporate a gating mechanism that allows them to retain information over extended sequences, making them well-suited to capture temporal dependencies in noisy and non-stationary financial time series. This capability is particularly important in stock markets, where delayed reactions, autocorrelations, and irregular patterns are common. By selectively preserving relevant past information through memory cells and forget gates, LSTMs have proven effective in a variety of forecasting tasks [311].

Similarly, **Gated Recurrent Units (GRUs)** offer a simplified gating structure by combining the forget and input gates into a single update gate, reducing computational complexity while preserving the ability to model long-term dependencies. GRUs have demonstrated comparable performance to LSTMs in several financial contexts, often with fewer parameters and faster training times [312].

In recent years, various extensions of LSTM-based models have been proposed to enhance forecasting accuracy in financial applications. For instance, Gosh et al. [313] employed a bidirectional LSTM (Bi-LSTM) to improve directional prediction in high-frequency trading environments, while Cao et al. [314] introduced a hybrid LSTM-CNN architecture to jointly capture local and sequential dependencies. Moreover, attention-based variants such as the Dual-Stage Attention RNN (DA-RNN) [315], as well as models integrating transformers with recurrent backbones, have shown improved performance in volatile markets [316].

Overall, LSTM and GRU architectures continue to play a central role in modeling temporal dependencies within multimodal financial prediction pipelines. Their ability to learn from sequential patterns makes them especially valuable when combined with other modalities, such as textual or relational data, or embedded within hybrid architectures.

Graph Neural Networks for Financial Forecasting

Graph Neural Networks (GNNs) [317] have emerged as a powerful paradigm for modeling structured relational data. In financial forecasting, they are particularly effective for capturing the complex inter-

dependencies among stocks, firms, or sectors—relationships that are often encoded as graphs based on price correlations, supply chain links, industry classifications, or co-movement patterns.

In multimodal contexts, GNNs are frequently used alongside other models to integrate structured market signals (e.g., historical prices) with unstructured information sources such as news articles or social media content [90]. Each node in the graph typically represents a financial entity (e.g., a stock), while node features may include numerical descriptors, textual embeddings, or multimodal vectors. By propagating information through graph edges, GNNs allow joint reasoning over individual stock attributes and their relational context [38, 318].

Formally, the financial market can be represented as a graph $G = (V, E)$, where V denotes the set of nodes (stocks) and E the set of edges (relations). Given initial node features $X \in \mathbb{R}^{|V| \times d}$, GNNs iteratively update the representation of each node v_i by aggregating information from its neighbors $\mathcal{N}(v_i)$, enabling the model to capture both local interactions and broader market structure.

Among the various GNN architectures, two have been particularly influential in financial applications: Graph Convolutional Networks (GCN) [85] and Graph Attention Networks (GAT) [319].

Graph Convolutional Networks (GCNs) extend the notion of convolution to graph-structured data by implementing localized message passing based on graph topology. In this framework, each node representation is iteratively updated by aggregating information from its neighbors, leveraging a normalized adjacency structure. This mechanism allows the network to learn expressive node features that capture both local and global structural patterns.

In the financial domain, GCNs have been applied to graphs constructed from stock correlation matrices [320] and sector-based hierarchies. Within multimodal architectures, GCNs offer a natural way to incorporate textual representations—such as embeddings extracted from financial news or earnings reports—into node attributes, while edge connections encode dynamic relationships between financial entities.

Graph Attention Networks (GATs) enhance GCNs by introducing an attention mechanism that enables the model to assign different levels of importance to neighboring nodes. This is particularly relevant in financial markets, where not all inter-stock connections are equally informative for price movements. Instead of uniformly averaging neighbor features, GATs learn to focus on the most influential relationships.

GATs have demonstrated strong performance in financial applications due to their ability to flexibly model heterogeneous dependencies. For instance, Goa et al. [321] introduce hierarchical GATs to capture interactions at both the sector and market levels, while other studies (e.g. [322, 320]) incorporate temporal information through dynamic graph modeling, allowing the representation of evolving financial structures.

Overall, GNNs provide a natural and flexible architecture for multimodal fusion. While Transformer-based models are effective at capturing modality-specific information (e.g., language or price patterns), GNNs excel at learning structured dependencies among financial entities. Recent work has begun integrating GNNs, particularly GATs, with other deep learning backbones such as LSTM or BERT, enabling joint learning over textual signals, temporal dynamics, and graph-structured relations [38].

14.2.4 Fusion Approach

In multimodal learning, fusion mechanisms refer to the strategies used to combine information from different input modalities—such as text, time series or graphs—into a unified representation suitable for prediction tasks. The choice of fusion strategy plays a crucial role in the model's ability to capture complementary information and exploit cross-modal interactions. Broadly, fusion mechanisms can be classified into three categories: early fusion, late fusion, and attention-based (or intermediate) fusion. Each of these strategies reflects a different design philosophy and offers specific trade-offs in terms of flexibility, complexity, and ability to model cross-modal interactions.

Early fusion

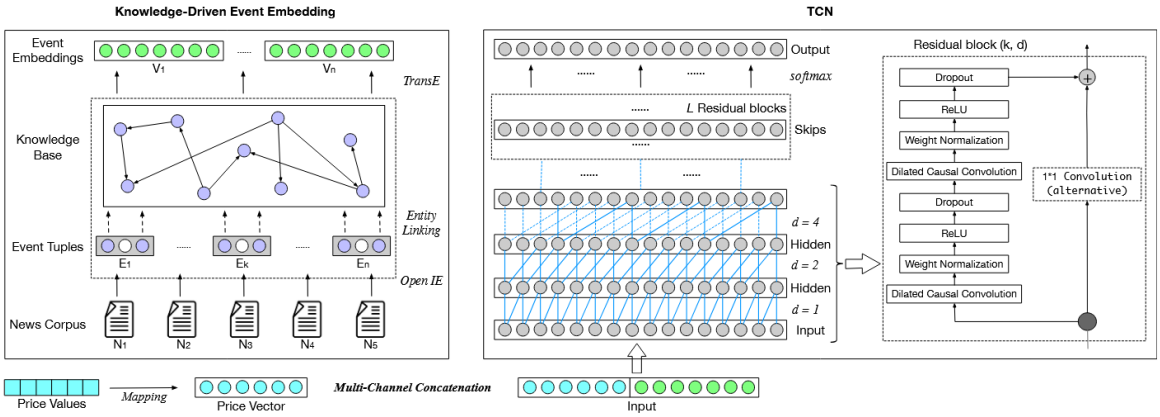


Figure 14.7: Example of early fusion from [36]. Following the vector representation of price values and events, the corresponding vectors are concatenated to produce a unified input to the KDTCN framework.

It refers to the combination of raw or low-level features from each modality into a single input representation, prior to any modeling. For example, one might concatenate a vector representing the sentiment of a news headline (extracted via a language model like BERT) with statistical indicators from financial time series (such as moving averages or volatility measures), and feed the resulting feature vector into a neural model like an LSTM or Transformer. This method allows the model to learn joint representations from the start and can capture interactions between modalities at the feature level. However, early fusion assumes that all modalities are temporally or structurally aligned, and may suffer when there are large discrepancies in scale, reliability, or noise. Additionally, when input dimensions are large, early fusion can lead to overfitting due to high model complexity.

Intermediate fusion

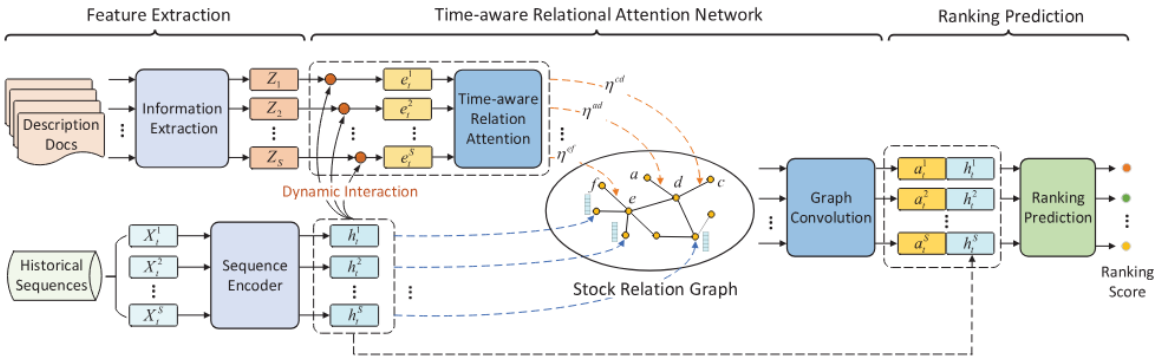


Figure 14.8: Example of intermediate fusion from [321]. The fusion of textual data and historical market data is performed through an attention mechanism to obtain representations enriched with relational features for all stock nodes.

Intermediate fusion provides a flexible and powerful mechanism to integrate different modalities during the representation learning phase. Rather than combining inputs at the feature level (as in early

fusion) or predictions at the output level (as in late fusion), intermediate-base fusion methods allow a modality to dynamically attend to another based on relevance, enabling fine-grained, context-dependent interactions. One of the most relevant intermediate fusion mechanisms is based on the *attention mechanism* [24], which computes a weighted sum over a set of values based on the similarity between a query and each key. Formally, given a set of queries $Q \in \mathbb{R}^{n_q \times d}$, keys $K \in \mathbb{R}^{n_k \times d}$, and values $V \in \mathbb{R}^{n_k \times d}$, the attention operation is defined as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V. \tag{14.2}$$

This allows the model to focus on the most relevant parts of one modality given the context of another. Attention-based fusion has been used in models that combine BERT-style textual encoders with LSTMs or GNNs that process temporal or relational data [38, 322]. Such models are particularly effective in financial settings, where modalities are often asynchronous, heterogeneous, and loosely aligned, and where capturing inter-modality dependencies is crucial for accurate predictions. Figure 14.8 illustrates an example of the intermediate attention fusion mechanism.

Late fusion

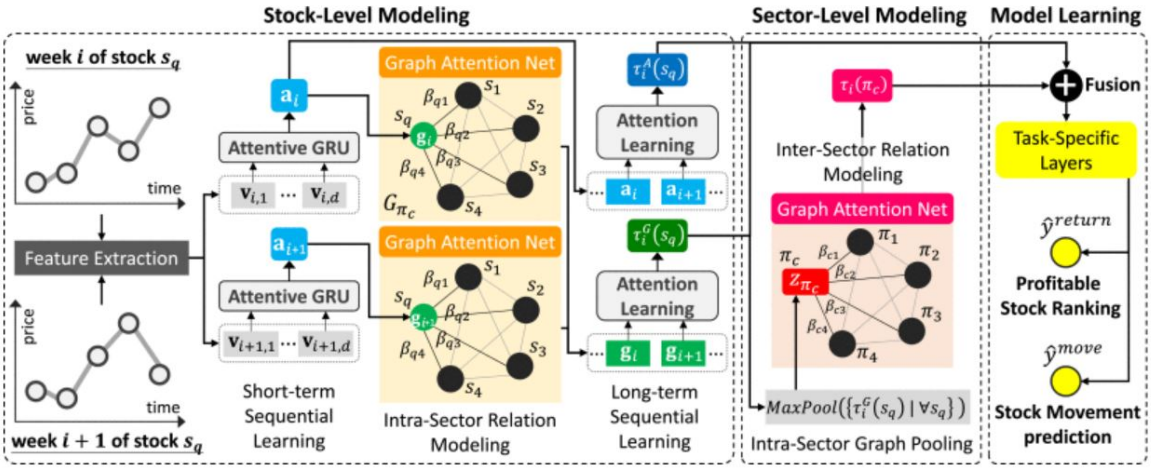


Figure 14.9: Example of late fusion from [318]. In this framework, the fusion between stock-level information and sector-level modeling is performed through the concatenation of feature embeddings, followed by a ReLU layer.

Late fusion, by contrast, delays the integration until after each modality has been processed by its own dedicated encoder. For instance, a graph neural network (GNN) might encode relational information between companies, while a Transformer processes news text, and a separate LSTM analyzes historical prices. Each of these models produces an output embedding or prediction, and these outputs are then combined—typically via averaging, weighted sum, voting, or a shallow classifier—into the final decision. This approach is modular and flexible: each branch can be optimized separately and can use modality-specific architectures.

In summary, early fusion captures low-level interactions but requires well-aligned inputs, late fusion provides modularity and robustness at the cost of cross-modal synergy, while intermediate fusion enables dynamic and contextual integration of modalities, and is especially powerful in domains like financial forecasting, where relationships between inputs are complex, nonlinear, and time-dependent.

14.2.5 Main Tasks

In financial forecasting, various modelling tasks can be defined based on what the system is trying to predict. These tasks are usually grouped into two main categories: *stock ranking* and *stock feature prediction*. Each task addresses a specific problem in financial decision-making and brings different advantages to investors, portfolio managers, and analysts.

Stock Feature Prediction

This category encompasses a range of tasks designed to predict specific quantitative indicators associated with individual stocks. It comprises several sub-tasks, each with different objectives and applications.

Stock Movement Prediction. This task is typically formulated as a classification problem. The goal is to predict whether the price of a stock will go up, down, or remain stable over a certain period. This simplifies the modeling process and allows the use of standard classification models. Despite being relatively coarse, this task is highly actionable for short-term trading signals.

Let P_t be the price of a stock at time t. The movement label y_t can be defined as:

$$y_t = \begin{cases} 1 & \text{if } P_{t+1} > P_t \\ 0 & \text{otherwise} \end{cases}. \tag{14.3}$$

Stock Return Prediction. In this task, the model predicts the future return of an asset, defined as: $r_{t+\Delta t} = \frac{P_{t+\Delta t} - P_t}{P_t}$. This is typically approached using regression methods trained to minimize the expected squared error. The output is a continuous-valued forecast, and the task is sensitive to outliers and heavy tails in the distribution of returns.

Volatility Prediction. Volatility forecasting involves predicting the future variability of asset returns. Accurate volatility estimates are essential for derivative pricing, risk assessment, and hedging. Models may output point forecasts (e.g. standard deviation) or distributional estimates.

Close Price Prediction. This task aims to estimate the actual closing price of a stock for the next trading session or over a future horizon. It is a regression task and can provide useful benchmarks for stop-loss levels or price targets. Formally, we can express this as: $\hat{P}_{t+1} = f_\theta(\mathcal{X}_t)$, where $\mathcal{X}_t$ represents features available at time t, and $\hat{P}_{t+1}$ is the predicted closing price.

Cross-Market Index Prediction. Cross-market index prediction deals with forecasting the values of broader market indices, such as the S&P500 or Nasdaq. These indices reflect the aggregated performance of multiple assets and are influenced by microeconomic and macroeconomic variables. Formally, this can be written as: $\hat{I}_{t+1} = f_\theta(S_t, M_t, T_t)$, where S_t represents stock-level features, M_t macroeconomic indicators, and T_t textual or event-based data.

Stock Ranking

Stock ranking aims to sort a list of stocks based on their expected performance, such as future return or risk-adjusted value. Unlike other tasks that try to predict exact prices or returns, stock ranking focuses on the relative order of the assets. This is especially useful in portfolio construction and decision support systems [38], where an investor might only be interested in the top-k performing stocks according to a model. In practice, this task assigns a score to each stock and produces a ranked list, enabling strategies like long-short trading or factor investing without relying on precise value predictions. Formally, given a set of stocks $\{s_1, s_2, \ldots, s_n\}$ and a predictive model f_θ, stock ranking seeks to compute a score $r_i = f_\theta(s_i)$ for each stock such that the ranked list $\{s_{(1)}, s_{(2)}, \ldots, s_{(n)}\}$ satisfies: $r_{(1)} \geq r_{(2)} \geq \cdots \geq r_{(n)}$.

Each of these tasks poses specific challenges in terms of data requirements, model interpretability, and performance evaluation. In recent years, multimodal architectures have been increasingly adopted to enrich input representations and improve forecasting outcomes across all of these task types.

14.3 A Taxonomy-Based Framework for Multimodal Stock Forecasting

To systematically organize this expanding body of literature, we propose a taxonomy-based framework that identifies and categorizes key dimensions along which existing works can be compared and analyzed. This taxonomy serves both as a conceptual map and as a tool for highlighting current trends, gaps, and future research opportunities in the field.

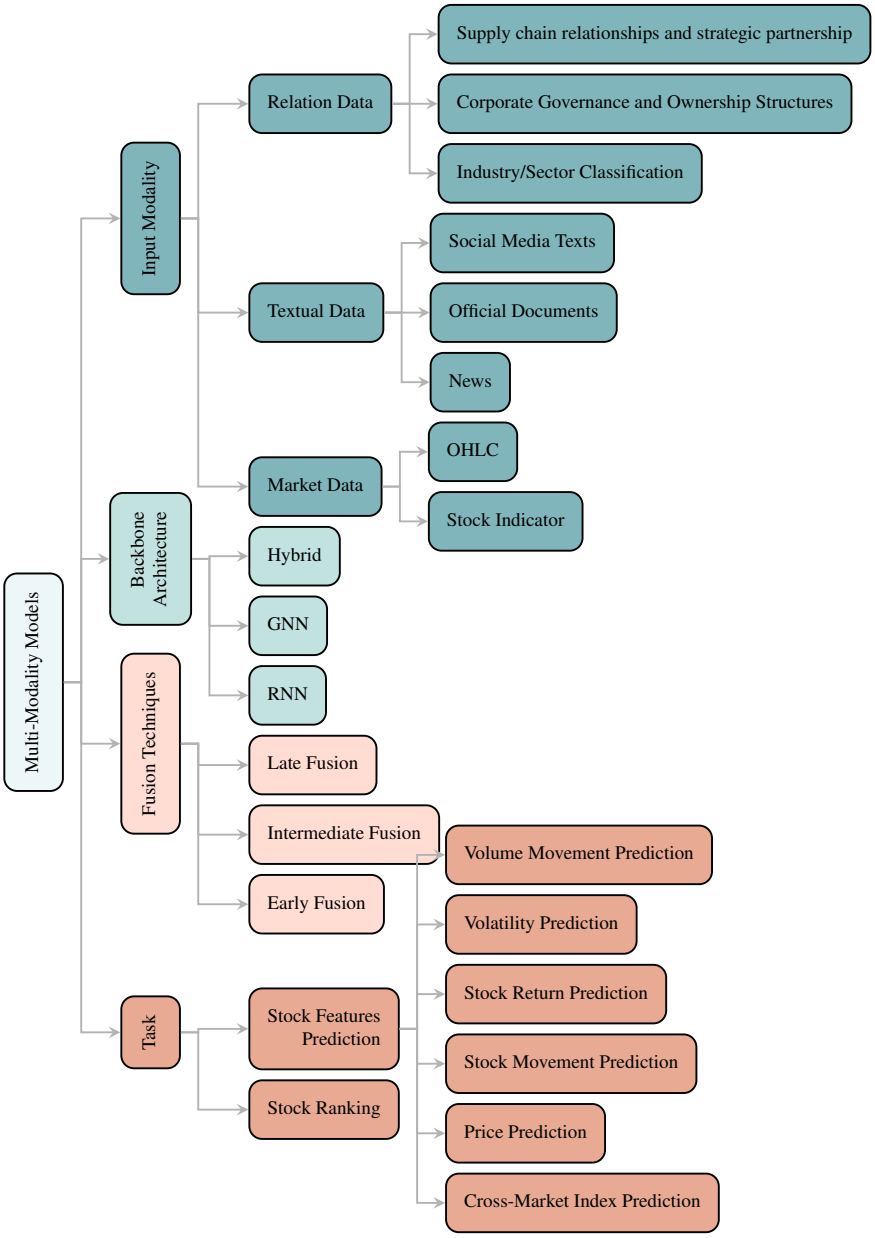


Figure 14.10: A taxonomy of Multi-Modality Models for financial prediction with their components and applications.

Figure 14.10 categorizes existing multimodal forecasting studies along four orthogonal dimensions, which correspond to those introduced in section 14.2.

The **Input-Modality** dimension specifies the data sources a model ingests. It spans: (i) market data such as OHLC price series and derived technical indicators; (ii) textual data drawn from social-media posts, official filings, and news articles; and (iii) relation data that encode firm–to-firm links—e.g., institutional ownership, governance structures, and industry or sector membership.

The **Backbone-Architecture** axis captures the neural family employed to process each modality. Most works rely on recurrent networks (LSTM/GRU) for temporal signals, graph neural networks (GCN/GAT and variants) for structured relational information, or hybrid stacks that combine multiple backbones to exploit complementary strengths.

The **Fusion-Technique** axis describes where and how modalities are integrated. Early fusion concatenates features before encoding, intermediate fusion merges latent representations through cross-modal attention or gating, and late fusion aggregates modality-specific predictions at the decision layer. This categorization highlights the trade-off between mutual information exchange and architectural modularity.

Finally, the **Task** dimension groups downstream objectives. Under stock-features prediction we include fine-grained targets such as volume movement, volatility, returns, directional movement, price levels, and cross-market index values; a separate branch, stock ranking, focuses on ordering assets (e.g., top-K selection) rather than forecasting point estimates. Together, these four dimensions provide a compact yet comprehensive map for positioning existing methods and identifying unexplored design combinations.

In the following subsections, we explore each dimension in more detail, providing representative examples from the literature and highlighting how different approaches position themselves within this taxonomy.

The following sections applies this taxonomy to analyze representative works in multimodal financial forecasting, enabling a structured comparison of their design choices and methodological trends.

14.3.1 Input Modalities

A central component in multimodal financial forecasting models is the selection and integration of diverse input modalities. Based on the literature, we identify three main categories: (i) market data, (ii) textual data, and (iii) relational or graph-based data.

Market data. Market data, primarily OHLCV time series enriched with technical indicators, serves as the foundational input for most quantitative forecasting models. For example, Wang et al. [323] propose a FFNN architecture enhanced by multimodal fusion strategies that include OHLC data. Similarly, Cheng et al. [290] combine market time series with graph-based relational structures to capture temporal and inter-asset dependencies. Another relevant work is Liu et al. [324], where the authors incorporate multiscale temporal data along with market prices into a GNN-based framework.

Textual data. Textual data, including news, reports, and social media, provides additional signals on sentiment, macro trends, and firm-specific events. In Li et al. [325], the authors integrate online news with stock price data through a multimodal LSTM architecture that captures the impact of events. Sawhney et al. [297] uses both tweets and news to extract time-sensitive semantic features using attention mechanisms. Similarly, Hu et al. [326] focuses entirely on news-driven predictions, emphasizing the chaotic but informative nature of financial news. Another example is Ang et al. [327], where the model integrates news at both local and global scales to jointly forecast multiple financial indicators.

Relational data. The third category comprises relational or graph-based data, capturing structured relationships among financial entities through static or dynamic graphs such as co-movement networks, knowledge graphs, or industry hierarchies. In Qian et al. [298], a multi-relational GNN captures heterogeneous relations (e.g., sector, supply chain, ownership) and temporal evolution. Similarly, Chang et al. [290] highlights how price time series can be enriched with topological relationships via GNN layers.

14.3.2 Model Architectures

The backbone architecture of a multimodal financial forecasting model plays a pivotal role in determining how effectively it can process and integrate different input modalities. Across the literature, we identify five main architectural categories: recurrent networks (LSTM/GRU), graph neural networks (GNNs), and hybrid architectures that combine multiple of the above.

RNN. Farimani et al. [293] propose an approach that aims to predict the absolute exchange rate price of an asset at the next trading hour by leveraging multimodal data: news content, aggregated mood sentiment, and market technical indicators. The model builds a latent concept space for news by clustering word embeddings and vectorizing news documents based on term frequencies of these concepts. Sentiment scores are derived from a fine-tuned financial transformer model (FinBERT [35]), while market data include common technical indicators normalized adaptively to adjust to non-stationary distributions. Feature extraction involves applying LSTM layers on news data, and recurrent layers on mood and market time series to capture temporal dependencies. An attention mechanism dynamically weighs the importance of news, mood, and market features over time. Finally, the fused features are passed through dense layers to predict a price change offset, which is added to the lagged average closing price to generate the final price prediction. The model is trained end-to-end using MSE loss optimized by Adam.

In Liu et al. [38], the authors propose a model for stock ranking that leverages both textual data and historical price information. The two modalities are processed independently: textual inputs are first embedded using BERT and then further refined through LDA-based feature extraction, while historical price data are directly encoded. Each modality is then passed through a separate LSTM network to capture temporal dynamics. Finally, the outputs are fused in a feed-forward layer and optimized using a pairwise ranking-aware loss function to produce a ranked list of stocks.

The framework proposed by Cheng et al. [328] predicts stock movements by integrating three types of information: technical indicators, news text, and inter-stock relationships. It first uses a tensor fusion module to combine technical and textual signals, capturing both their individual and joint effects. The resulting fused vectors, representing daily firm-level information, are processed by a GRU to model temporal dynamics and generate sequential embeddings. To capture the influence between firms—referred to as momentum spillover the model constructs relational embeddings using a dynamic graph at each time step based on the correlations among stocks. In this way, the model dynamically learns the strength of inter-firm relations based on market signals, guided by attention mechanisms that account for both node attributes and learned graph structures. Finally, sequential and relational embeddings are concatenated and passed to a prediction module to classify the future stock movement direction. The entire model is trained end-to-end using a cross-entropy loss.

Zao et al. [299] propose a model that predicts stock movements by learning sequential embeddings through a GRU that processes fused multimodal market signals, combining technical indicators and news sentiment over time. This captures the temporal dynamics of each stock. Relational embeddings are derived from a bi-typed market knowledge graph linking companies and executives, using a Dual Attention Network to model both inter-class and intra-class interactions via hierarchical attention mechanisms. The sequential embeddings from the GRU and the relational embeddings are then combined and passed through a feed-forward network with softmax for prediction. The entire system is trained end-to-end with cross-entropy loss.

The PEN architecture [329] primarily relies on recurrent neural networks, specifically LSTM and GRU. It uses a bidirectional GRU layer to generate low-dimensional text embeddings from daily news. A Text Memory Unit, inspired by LSTM, preserves important textual information over time. Text and price data are combined through recurrent modules to learn shared representations. A recurrent variational autoencoder based on GRU is then used to generate stock movement predictions. Finally, a temporal attention mechanism aggregates predictions across time steps for improved accuracy. Over-

all, PEN focuses on LSTM and GRU to capture temporal dynamics in both text and price data, without employing graph neural networks.

GNN. Pei et al. [320] propose a framework that jointly captures temporal evolution and static relationships among stocks using a dual-graph approach. The model constructs two dynamic graphs: a temporal graph that changes over time to reflect evolving market behavior, and a static relational graph capturing long-term dependencies. To extract representations, the authors employ a GCN with contrastive learning to align node embeddings across time steps, encouraging consistency while allowing for temporal variation. This dual perspective enhances the model's ability to represent both persistent structures and dynamic patterns in financial data.

Kumar et al. [330] present a model that uses dynamic graphs to reflect the time-varying relationships among financial assets. The architecture incorporates temporal snapshots of asset interactions, which are fed into a GAT to model both the evolution of the graph and the time series. The GAT module learn context-aware features from each temporal graph, effectively capturing volatility-relevant patterns across changing market structures.

In Sawhney et al. [331], the authors propose a framework for stock movement prediction that integrates both tweets and historical price data. After encoding the two modalities into feature vectors, these are used as node attributes in a Graph Attention Network (GAT), where each node represents a stock. The GAT is then trained to perform node classification, predicting the binary movement (UP or DOWN) of each stock based on its features and its relationships with other stocks.

In Xu et al. [332], the authors present a GCN-based approach for stock movement prediction that leverages event information related to individual stocks. The core intuition is that financial events can significantly influence stock behavior, and modeling their relational structure can enhance predictive performance. The model uses a Graph Convolutional Network (GCN) to capture the importance of each event by learning from the event-stock interaction graph. For each stock, the representations of all associated events are concatenated, and the resulting vector is passed through a dense layer to generate the final prediction.

Feng et al. [333] use LSTM networks combined with an attention mechanism to extract a feature matrix from various technical indicators (such as Exponential Moving Average, Average Price, etc.). This feature matrix is then used as the input representation for the nodes in a stock relation graph. A Graph Attention Network (GAT) is applied to this graph to capture inter-stock dependencies, and the final output is obtained through a prediction layer optimized using a point-wise regression loss objective.

Gao et al. [321] propose TRAN, a framework that combines market data with features extracted from text for stock ranking prediction. Specifically, market data are used as node features in a graph, while an attention mechanism is employed to integrate textual and numerical information. The resulting attention scores are used as edge weights to represent the strength of relationships between stocks. A Graph Convolutional Network (GCN) is then applied to this graph to generate final stock rankings.

Qian et al. [298] introduce a sophisticated GAN framework that integrates multiple types of financial relations (e.g., industry, supply chain, co-movement) into a dynamic graph structure. The model combines relational-aware graph attention with temporal encoding, allowing it to simultaneously process heterogeneous relations and their temporal evolution. The GAN architecture includes multi-relational attention layers that weigh the importance of each edge type and time step, enabling a comprehensive and dynamic understanding of stock interactions for investment prediction.

Hybrid. These architectures typically integrate different neural modules—such as GNNs, LSTMs, and attention mechanisms—into a cohesive system that can model complex interactions across time, modalities, and entities.

Hsu et al. [318] introduce a hybrid GRU-GAT framework for ranking stocks based on their predicted return ratios and finding future stocks with positive movements. Starting from historical price

data, GRU layers are used to extract temporal features, which are then incorporated into stock-level and sector-level interaction graphs. These graphs are encoded using multi-head attention layers, and the resulting node representations are optimized via task-specific loss functions designed to guide stock ranking and identify assets with positive future movements.

Zhao et al. [334] proposed a hybrid model that combines GCN and LSTM to capture both relational and temporal patterns in financial data. It integrates long-term correlations between stocks through a multi-relational stock graph built from historical price and volume data. For short-term modeling, separate GCNs are applied to price and volume graphs, followed by a graph-aware LSTM that captures temporal dependencies within each stock. News events are encoded via an attention-based LSTM and fused with stock features at the final time step, enriching the representation. Additionally, it uses a Siamese network to project price and volume representations into a shared space and optimizes their correlation. The final prediction is made through a fully connected layer.

Cheng et al. [306] propose a hybrid architecture that combines GCN and LSTM-based models to learn rich event embeddings for quantitative investment. It integrates three components: (1) a relation extraction module using BiLSTM with attention to detect entity relations from financial news; (2) an event representation module that uses a Neural Tensor Network (NTN) and GCNs over a financial knowledge graph (FinKG) to capture structured and contextual information; and (3) a multi-source attention mechanism to fuse embeddings from events, relations, and the graph. This GCN-LSTM hybrid framework captures both direct and lead-lag effects of events on stock movements.

Lazcano et al. [303] propose an architecture for time series prediction that is a hybrid model called BiLSTM-GCN that combines two neural network types to leverage their complementary strengths. The core architecture is a Graph Convolutional Network combined with LSTM (GCN-LSTM). This model constructs a graph from the correlation matrix of the time series data, enabling it to capture spatial relationships among nodes through graph convolutional layers. The spatial features extracted by the GCN are then processed by LSTM layers to model temporal dynamics. Dropout and dense layers are added to improve generalization and performance. This hybrid architecture, named BiLSTM-GCN, effectively integrates the temporal modeling capabilities of the BiLSTM with the spatial-temporal learning of the GCN-LSTM, leading to improved prediction accuracy.

The architecture proposed in Li et al. [335] is a hybrid model combining LSTM and GCN to predict overnight stock movements. News headlines are encoded with LSTM and attention, merged with stock embeddings, and used as node features in a stock correlation graph. The graph uses RGCN layers to capture positive and negative stock relationships. LSTM units between GCN layers help avoid oversmoothing by dynamically gating information. A global node models overall market trends. The final node representations are used for binary stock movement prediction.

The ST-Trader model [336] tackles stock price prediction by capturing both spatial and temporal dependencies. First, a Variational Autoencoder (VAE) is used to encode static fundamental features of each stock into a low-dimensional latent space. Based on the distances between these latent vectors, a spatial graph is constructed, where nodes represent stocks and edges reflect their learned similarity. To model temporal dynamics, the framework employs a hybrid GCN-LSTM architecture: Graph Convolutional Networks apply spectral filters (via Chebyshev polynomials) to propagate information across the graph, while LSTM units capture sequential patterns in historical stock prices. By integrating graph-based spatial structure with time-series modeling, ST-Trader generates more informed predictions of future stock prices.

Liu et al. [294] propose a hybrid architecture that combines multiple deep learning techniques to integrate market, text, and social data for stock return prediction. It uses LSTM for time series market data, BERT for encoding financial text, and graph neural networks (GNN) to capture social network relationships. These diverse feature representations are fused through a multimodal attention mechanism, allowing the model to adaptively weigh each data source. Finally, classic machine learning methods like XGBoost and LASSO select the most predictive factors. This hybrid approach effectively leverages different data types and modeling techniques to improve factor robustness and

prediction accuracy.

This diversity of architectures reflects the multifaceted nature of financial forecasting tasks and the need for models that can flexibly adapt to heterogeneous data sources. The ongoing shift toward transformer and graph-based models, and especially their hybrid combinations, signals a promising direction for future work.

14.3.3 Fusion Approach

Early Fusion. Early fusion techniques combine data from multiple sources at the input level, before being processed by the model. This approach concatenates or otherwise merges raw or low-level features (e.g., prices, volumes, textual indicators) into a single input vector that is fed into the model. The advantage is that the model learns from all modalities simultaneously from the very beginning, potentially capturing simple correlations across data types.

Cheng et al. [328] propose an early fusion strategy using a Tensor Fusion Module to jointly model numerical and textual features for stock prediction. For each stock, technical indicators and textual embeddings (e.g., from news articles) are first extracted and then fused through a bilinear tensor product that captures cross-modal interactions. Additionally, a linear transformation of the concatenated features preserves their independent contributions. The final fused representation is obtained by combining both terms and applying a non-linear activation function. This approach enables the model to learn task-specific and stock-specific feature interactions in a unified representation.

In Li et al. [335], the authors construct a stock correlation graph where each node represents a stock and edges reflect positive or negative correlations based on historical prices. Each node is enriched with news text, encoded via an LSTM followed by an attention mechanism that uses the stock embedding as a query. The resulting text representation is concatenated with the stock embedding to form the node features. This multimodal graph, combining market and textual signals, is then processed by a hybrid LSTM-RGCN model to predict stock movements.

Deng et al.[36] propose an early fusion approach to represent structured event information extracted from financial news. The method starts by applying Open Information Extraction (OpenIE) [337] to transform unstructured news text into structured event tuples, each consisting of a subject, predicate, and object. These components are then linked to a Knowledge Graph to enrich them with semantic and relational information. To capture broader contextual knowledge, the model also includes the immediate neighbors of the linked entities in the graph. For each element of the event tuple, three types of features are extracted: embeddings from the Knowledge Graph, contextual embeddings from neighboring entities, and standard word embeddings. All these representations are then concatenated into a single vector. This fused representation is used as input for the stock movement prediction tasks.

Xu et al. [332] propose a multi-step fusion mechanism that integrates event text and stock-specific historical context. First, events extracted from financial news are encoded using a type-specific multi-head attention mechanism, which highlights the most relevant tokens based on the event type. These daily event embeddings are then processed by an LSTM to capture temporal dependencies over recent days. In parallel, the model encodes each stock's historical context by processing past events and their market feedbacks (e.g., price or volume changes) with two LSTMs, whose outputs are concatenated. To model the stock-dependent effect of event information, the event embedding and the stock context embedding are fused via a feed-forward network. This produces a stock-specific modulation of the event signal, enabling the model to capture how different stocks react differently to similar events.

Zhao et al. [299] present a model that performs early fusion of multi-modal time-series signals, specifically combining technical indicators and sentiment features derived from news. The authors note that stock prices are influenced by market signals over multiple days, so their model takes into account the historical sequence of fused features when predicting future movements. The fusion mechanism is based on a Neural Tensor Network, which captures the complex interactions between

the two modalities—technical and sentiment—by modeling their relationships across multiple dimensions. The resulting fused representation for each day is then passed through a GRU layer, and the final hidden state is used to encode the temporal dynamics of each stock.

MAN-SF: Multipronged Attention Network for Stock Forecasting, proposed by Sawhney et al. [331], combines historical price data and social media signals using a generalized bilinear fusion mechanism, which explicitly models pairwise interactions between the two modalities. Price features are extracted using a GRU followed by temporal attention, which highlights the most informative trading days. Social media information is encoded hierarchically: a GRU with intra-day attention identifies impactful tweets within each day, and a second GRU with temporal attention aggregates information across days. The resulting price and tweet representations are then fused through a bilinear transformation, which captures their joint dynamics more effectively than simple concatenation or averaging. This fusion layer enables the model to learn richer cross-modal interactions, leading to better market trend prediction.

Intermediate Fusion. Intermediate fusion techniques integrate data from multiple sources within the model during training. Unlike early fusion, which combines raw inputs before processing, or late fusion, which merges separate model outputs, intermediate fusion merges features at hidden layers. This allows the model to learn joint representations that capture complex interactions between different data modalities, improving performance on tasks where multiple types of information must be considered together to produce a single final output.

DGRCL proposed in Pei et al. [320] uses an intermediate fusion strategy to combine temporal stock data and structural company relations. It first constructs a dynamic graph at each time step by computing pairwise stock similarities using Dynamic Time Warping (DTW) [338]. Then, it generates initial node embeddings that capture both stock-specific temporal trends and relational constraints derived from company links. These enhanced embeddings are passed through an RNN to capture sequential dependencies and produce final node representations. The fusion of modalities occurs within the RNN, allowing the model to jointly learn from both temporal patterns and structural relations before making predictions.

The Melody-GCN model [324] combines stock price data and media text information through an intermediate fusion strategy within its processing pipeline. Initially, the model separately analyzes historical price movements and text embeds derived from media sources, extracting features from each modality along multiple temporal scales. Crucially, it does not merge these two data types at the input or output stages. Instead, it fuses them during feature extraction: the encoded price and text representations are combined midway, processed jointly, and then used to enhance each other. This allows price-based insights to inform text understanding, and vice versa, creating richer, context-aware representations.

In the Dandelion model [339], intermediate fusion occurs after each modality (such as time series, fundamentals, and graphs) is processed by a dedicated encoder. The resulting latent representations are then integrated through a multimodal attention mechanism, which dynamically weighs the contribution of each modality based on its relevance to the task. To encourage alignment across modalities, a regularization term is applied to minimize their distance in the shared latent space. This fusion step takes place before prediction and enables selective information sharing across related tasks.

Hu et al. [326] propose a stock trend prediction model that performs intermediate fusion of textual data. Individual news articles for each day are first converted into vectors and weighted through an attention mechanism to highlight the most relevant ones. These daily aggregated vectors are then combined along the temporal sequence using another attention layer that selects the most influential days. Thus, fusion occurs in two stages—at the news level and at the temporal level—gradually integrating textual information into the final representation used for prediction.

Prediction-Explanation Network (PEN) for stock movement prediction, proposed by Li et al. [329], performs intermediate fusion of text and price data through its Shared Representation Learning (SRL)

module. SRL integrates daily text embeddings and historical price features by first selecting relevant texts via a Text Selection Unit (TSU) that assigns importance scores to each text. Then, a Text Memory Unit (TMU), inspired by LSTM, preserves important textual information over time. Finally, the Information Fusion Unit (IFU) merges the text memory and price data into a shared hidden representation, which captures their interaction and is recurrently updated across time steps. This fused representation is used downstream for variational inference and prediction, enabling the model to jointly leverage textual and price signals in a temporally-aware manner.

Shi et al. [340] propose a model that leverages four types of graphs representing different stock relations (industry/region, concept, volatility) and uses GCN to extract temporal embeddings that capture both relational and time-varying information. These embeddings are then fed into an LSTM to predict stock price movements. To enhance prediction performance, embeddings from the different graphs are fused using three strategies: concatenation, mean-filtering, and max-filtering. These fusion methods enable efficient integration of multiple knowledge sources, balancing between increased embedding dimensionality (concatenation) and information compression (filtering), thereby improving the model's predictive power.

The model proposed in Feng et al. [333] fuses temporal and spatial financial data using an intermediate fusion mechanism within a unified architecture. It processes stock time series with an attention-based LSTM to capture key temporal features and extracts spatial relationships from stock correlations via a stacked graph neural network combining graph convolution and graph attention layers. These two feature sets—temporal embeddings and graph-based spatial embeddings—are integrated during training through joint optimization, allowing the model to learn complex interactions between stock trends over time and their inter-stock correlations for improved stock ranking predictions.

Late Fusion. Late fusion techniques combine the outputs of separate models, each trained on a different data modality. In this approach, each model processes its input independently and produces its own prediction or feature representation. These outputs are then aggregated—using methods like averaging, weighted voting, or a meta-learner—to produce the final prediction. Late fusion is particularly useful when modalities are highly heterogeneous or asynchronously available, and it allows leveraging specialized models for each data source.

Liu et al. [294] adopts a late fusion approach to predict stock returns by first independently learning feature representations from multiple modalities—market data (via LSTM/Transformer), textual data (via BERT), and social data (via GNN). Each modality is processed by a dedicated model, and their outputs are later combined using a multimodal attention mechanism, which assigns adaptive weights to each modality. The resulting fused representation is then passed to a factor selection module (e.g., XGBoost or LASSO) to extract the most predictive features.

The fusion mechanism in Lazcano et al. [303] follows a late fusion approach, where two separate models—a BiLSTM and a GCN-LSTM—are first pretrained independently on the time series data. Their outputs are then concatenated to form a combined feature vector, which is passed through additional dense layers to integrate the complementary information extracted by each model. This approach allows the hybrid network to leverage both temporal dependencies captured by the BiLSTM and spatial-temporal correlations modeled by the GCN-LSTM, resulting in improved prediction accuracy.

The model proposed in Xiang et al. [341] employs a late fusion strategy to combine temporal and relational information. Specifically, it first encodes each stock's historical price sequence and its graph-based neighborhood separately. Temporal representations are obtained using a transformer, while relational features are extracted through a heterogeneous graph attention mechanism over dynamic correlation-based graphs. Late fusion occurs after these independent encodings: an attention mechanism integrates the stock's own features with those from positively and negatively correlated neighbors. This late-stage combination allows the model to preserve the specific strengths of temporal and structural signals before merging them into a unified representation for prediction.

In Sawhney et al. [342], the fusion of different data sources is achieved through an ensemble approach that combines outputs from separate models trained on distinct data types: the textual analysis of earnings calls and the historical stock price data. Historical stock prices over the 30 days before an earnings call are modeled using Support Vector methods—Support Vector Regression for volatility and Support Vector Classification for price movement. These predictions are then combined with outputs from models processing textual and aligned speech features from the earnings calls. To optimize this fusion, the ensemble weights assigned to each component are carefully tuned by experimenting with different values in small increments, allowing the model to find the best balance between market data and earnings call information for improved prediction performance.

Zao et al.[334] predict trading volume movement by combining long-term stock relations, short-term price and volume fluctuations, and sudden financial news. It builds a graph connecting stocks based on historical price and volume correlations, capturing different types of relationships. Price and volume data are processed separately through graph neural networks and a specialized LSTM to model temporal changes while reducing noise. Sudden events from news headlines are encoded and fused by adding them as the last time step in the sequence, creating a combined temporal graph that integrates news with transaction data. The fusion happens by merging stock embeddings, encoded news features, and processed price and volume vectors into a single representation used for final prediction. This approach allows the model to effectively combine diverse data sources for better volume movement forecasting.

In RGStockNet [343], the fusion step integrates two complementary representations to improve stock trend prediction. The first is the relational embedding, which captures structural knowledge about inter-stock relationships learned from a knowledge graph built on pairwise time series interactions. This embedding provides a global market context for each company. The second is the time series embedding, obtained from a dedicated encoder that processes the recent historical data of the target stock, such as daily prices and other numerical indicators. This representation reflects the local temporal dynamics of the individual stock. Fusion is performed by a fully connected network which outputs the probability distribution over the possible future trends.

14.3.4 Predictive task

Stock movement prediction. Stock movement prediction focuses on the directional change in the price of a stock over a given time horizon. Many studies formalise the task as a classification (binary or multiclass) to identify whether a stock will go up, down or remain stable. For example, in Li et al. [325], a fusion of financial news and market data is used to predict binary price movements. Similarly, Xiang et al. [341] integrate temporal market data and graph structures to improve the performance of binary classification. Other works such as Pei et al. [320], Cheng et al. [306] and Ye et al. [344] similarly frame the forecasting task as binary classification, exploiting graph representations and multimodal inputs to capture complex dependencies. In Cheng et al. [290], the assignment of labels is determined using two thresholds: an upper threshold r_{up} and a lower threshold r_{down}. If the return R_r is greater than or equal to r_{up}, the movement is classified as UP; if it is less than or equal to r_{down}, it is classified as DOWN. When the return falls between these two thresholds, it is considered NEUTRAL. This task remains a dominant choice in the literature due to its interpretability, ease of labeling, and practical relevance in algorithmic trading applications.

Stock return. This task aims to estimate the percentage change in a stock's price over a specified time period, providing insight into potential future gains or losses. It is often modeled as a regression problem. In Wang et al. [323], by introducing adaptive feature selection modules, multimodal fusion modules, and regularization strategies, the proposed model can dynamically adjust feature weights, integrate multiple modality information (such as historical prices, technical indicators, and market sentiment), and effectively suppress overfitting problems. Zuou et al. [339] propose a time series

forecasting framework named Dandelion, which leverages multiple modalities for the prediction.

Cross-market index prediction. It addresses the forecasting of foreign or international stock indices by leveraging multimodal signals. Lee et al. [345] exemplifies this strategy. The authors developed stock prediction models that combine information from the South Korean and US stock markets by using multimodal deep learning.

Stock ranking. In financial forecasting, stock ranking is a task that goes beyond binary or multi-class classification by aiming to order a set of stocks based on a specific target signal, such as expected return, profitability, or investment potential. Rather than predicting individual labels for each stock, the goal is to generate a relative ranking that reflects the most promising investment opportunities. This setup is particularly useful in portfolio construction or recommendation systems, where the interest lies in selecting the top-K performing stocks rather than predicting the behavior of each in isolation. Liu et al. [38] focuses on exploiting the relationships among stocks through multi-modal data and graph-based learning methods to solve the stock rank prediction. They propose a Multi-modal Temporal Dynamic Graph method (MTDGraph) with the point-wise regression loss and ranking-aware loss to obtain the appropriate stock rank list. Sawhney et al. [297] propose a hierarchical approach to learning-to-rank that uses textual data to make time-sensitive predictions to rank stocks based on expected profit.In Liu et al. [38], textual features extracted from documents using LDA are combined with historical market data. These two modalities are processed independently and then fused in a fully connected layer to predict a ranking of stocks. Lastly, Gao et al. [321] constructs a graph of the stock relation and extracts features from stocks and documents for the recommendation of the stock according to the ranking of the return ratio.

Close price prediction. The objective of close price prediction is to forecast the actual future price of a stock rather than its return. Farimani et al. [293] employ a hybrid model for price regression that combines different components (FinBERT, CNN and LSTM).

Volatility prediction. It aims to estimate the expected variability of a stock's returns, a critical component for risk-sensitive applications. For example, Ibrahim et al. [346] introduced the model DynGWN that relies on a Tensor Graph Convolutional Module (TGCM), which captures dynamic trends in graphs effectively in the time-varying graph representations.

Multi-task learning. A single model is trained to predict multiple financial variables simultaneously, such as stock movement, volatility, or return. This approach is motivated by the fact that these tasks are often correlated: for example, volatility patterns can provide valuable context for predicting stock direction or return. Architecturally, multi-task models typically share the same feature extraction and fusion layers but add task-specific output heads, each corresponding to a distinct predictive objective. In Sawhney et al. [342], the authors propose a multitask approach that uses textual features and attentive audio alignment to predict the movement and volatility of the stock price simultaneously. Similarly, Ang et al. [327] propose Guided Attention Multimodal Multi-task Network (GAME), a model that captures both global and local multimodality information for return and volatility predictions.

Table 14.1 summarizes a collection of works in multimodal financial forecasting, comparing the input modalities used, model architectures, fusion strategies, and prediction tasks. In this context, the market modality refers to all numerical features directly related to the stock, in addition to the target variable (e.g., future return or price movement). Typical examples include historical prices, technical indicators (such as moving averages or RSI), trading volume, and volatility measures.

Table 14.1: Overview of analysis dimensions, detailing input modalities, model architectures, fusion strategies (early $\oplus$, intermediate $\odot$, late $\cup$), and prediction tasks (cross-market index prediction, price prediction, stock movement prediction, stock ranking, stock return prediction, volatility prediction, volume movement prediction).

Paper	Input modality			Architecture	Fusion approach	Task
	Market	Textual	Relation			
Li et al. [325]	✓	✓		RNN	$\oplus$	
Liu et al. [294]	✓	✓	✓	Hybrid	$\cup$	
Lee et al. [345]	✓			RNN	$\oplus$ $\odot$ $\cup$	
Farinami et al. [293]	✓	✓		RNN	$\cup$	
Wang et al. [323]	✓		✓	FNN	$\cup$	
Cheng et al. [290]	✓	✓	✓	GNN	$\oplus$	
Zhou et al. [339]	✓	✓	✓	Hybrid	$\odot$	
Xiang et al. [341]			✓	GNN	$\cup$	
Ibrahim et al. [346]			✓	Hybrid	$\cup$	
Pei et al. [320]	✓		✓	GNN	$\odot$	
Kumar et al. [330]			✓	GNN	$\odot$	
Hsu et al. [318]			✓	Hybrid	$\cup$	
Lazcano et al. [303]			✓	Hybrid	$\cup$	
Ang et al. [327]		✓	✓	GNN	$\cup$	
Sawhney et al. [342]		✓		RNN	$\cup$	
Liu et al. [324]	✓	✓		GNN	$\odot$	
Zhao et al. [334]		✓	✓	Hybrid	$\cup$	
Ye et al. [344]	✓		✓	Hybrid	$\odot$	
Liu et al. [38]		✓	✓	RNN	$\cup$	
Cheng et al. [306]	✓	✓	✓	Hybrid	$\odot$	
Cheng et al. [328]		✓	✓	RNN	$\oplus$	
Sawhney et al. [331]		✓		GNN	$\oplus$	
Li et al. [335]		✓		Hybrid	$\oplus$	
Hu et al. [326]	✓	✓		RNN	$\odot$	
Sawhney et al. [297]		✓		RNN	$\cup$	
Li et al. [329]		✓		RNN	$\odot$	
Deng et al. [36]		✓	✓	RNN	$\oplus$	
Qian et al. [298]	✓		✓	GNN	$\odot$	
Shi et al. [340]	✓		✓	Hybrid	$\odot$	
Li et al. [343]	✓		✓	Hybrid	$\cup$	
Zhao et al. [299]	✓	✓		GRU	$\oplus$	
Hou et al. [336]			✓	Hybrid	$\oplus$	
Xu et al. [332]	✓		✓	GNN	$\oplus$	
Feng et al. [333]	✓		✓	GNN	$\odot$	
Gao et al. [321]	✓	✓	✓	GNN	$\odot$	

14.4 Evaluation and Comparative Analysis

14.4.1 Data source and Datasets

Market data. Most of the works reviewed in this survey rely on financial market data obtained from Yahoo Finance[3], a widely used open-access platform that provides historical and real-time data

[3] https://finance.yahoo.com/

on stock prices, volumes, indices, and fundamental indicators (e.g., P/E ratios, dividends, earnings). The main strengths of Yahoo Finance include its accessibility, the breadth of data coverage, and the simplicity of use through both its web interface and APIs (such as `yfinance`[4] in Python). However, it has some limitations: the data may be subject to occasional inconsistencies or missing entries, especially in high-frequency settings or for less-liquid assets.

Textual Information. Textual data employed in multimodal financial prediction tasks are typically sourced either from proprietary platforms or publicly available datasets. A substantial body of research relies on financial news articles and reports disseminated by major media outlets such as *Reuters*[5], *Bloomberg*[6], and *MarketWatch*[7]. These sources offer high-quality, timely, and domain-specific content that is instrumental in capturing market sentiment and detecting event-driven signals. In certain instances, access to such data is facilitated through institutional partnerships or previously compiled corpora—for example, the dataset introduced by Duang et al. [31], which includes Reuters and Bloomberg articles collected between October 2006 and December 2015. In other instances, publicly available datasets are utilized to promote reproducibility and facilitate model comparability. The *CMIN-US* dataset [32] comprises extensive financial texts alongside stock price time series data from both the U.S. and Chinese markets. Another widely adopted resource is the *ACL18* dataset [33], which integrates two modalities: social media data and historical price information. It includes 387,045 tweets related to 70 companies spanning seven industries. The *2017 Earnings Conference Calls dataset* [34] provides transcripts of earnings calls sourced from *Seeking Alpha*[8], annotated with metadata such as speaker identities and their corresponding speech content.

14.4.2 Evaluation Metrics in Multimodal Financial Forecasting

To evaluate multimodal models in financial forecasting, studies employ diverse metrics spanning classification, regression, ranking, and financial performance indicators [290, 320, 324, 36]. In classification tasks, metrics such as accuracy, precision, recall, and F1-score are common, though class imbalance can distort results. The Matthews Correlation Coefficient (MCC) [347] is therefore often preferred, as it balances true and false predictions and remains robust under imbalance. For regression tasks, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are widely applied [294, 323, 293], with Median Absolute Percentage Error (MdAPE) offering resilience against outliers.

In stock ranking and portfolio allocation, ranking metrics are essential. Mean Reciprocal Rank (MRR@k) is used to evaluate the identification of top-performing assets [318, 38], complemented by Precision@k and Recall@k, which respectively assess accuracy and coverage within the top-k predictions. Together, these measures capture the prioritisation of profitable opportunities.

Overall, the choice of metrics reflects the heterogeneity of tasks, from movement classification to ranking and portfolio optimisation. While standard metrics provide comparability, specialised ones like MCC and MRR@k yield deeper insights into model behaviour and investment relevance, making metric selection critical to avoid misleading conclusions.

14.4.3 Analysed Markets

Table 14.2 summarises the markets and indices analysed across the reviewed studies, a factor that critically shapes input data, prediction dynamics, and model generalisation. Most works focus on

[4] `https://pypi.org/project/yfinance/`
[5] `https://www.reuters.com/`
[6] `https://www.bloomberg.com`
[7] `https://www.marketwatch.com/`
[8] `https://seekingalpha.com/`

Table 14.2: Overview of additional characteristics of the analysed papers, including performance metrics (e.g., Accuracy (A), Precision (P), Recall (R), F_1-score (F_1), Information Ratio (IR), and Investment Return Ratio (IRR)); dataset accessibility (private 🔒 vs. public 🔓); country of origin (China 🇨🇳, Japan 🇯🇵, South Korea 🇰🇷, Taiwan 🇹🇼, U.S. 🇺🇸, and multi-country datasets 🌍); and whether the approach has been applied in real-world scenarios.

Paper	Metrics	Dataset Access			Country	Real-world application
		Market	Textual	Relation		
Li et al. [325]	**A** P R **F_1** AUC IR IRR MAE MAPE **MCC** MdAPE MRR MSE R^2 RMSE	🔓	🔓		🇨🇳	✓
Liu et al. [294]	**A** P R **F_1** AUC IR IRR MAE MAPE MCC MdAPE MRR **MSE** **R^2** RMSE	🔓	🔒	🔓	🇺🇸	
Lee et al. [345]	A P R F_1 AUC IR IRR MAE MAPE MCC MdAPE MRR **MSE** R^2 RMSE	🔓			🇰🇷 🇺🇸	
Farinami et al. [293]	A P R F_1 AUC IR IRR MAE MAPE MCC MdAPE MRR **MSE** R^2 RMSE	🔓	🔓		🇺🇸	
Wang et al. [323]	A P R F_1 AUC IR IRR **MAE** MAPE MCC **MdAPE** MRR **MSE** R^2 **RMSE**	🔓		🔓	🇺🇸	
Cheng et al. [290]	A P R **F_1** AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE	🔓	🔓	🔓	🇨🇳	✓
Zhou et al. [339]	**A** P R F_1 AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE	🔓	🔓	🔓	🇺🇸	
Xiang et al. [341]	**A** P R F_1 AUC **IR** IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE			🔓	🇺🇸 🇨🇳	✓
Ibrahim et al. [346]	A P R F_1 AUC IR IRR **MAE** MAPE MCC MdAPE MRR MSE R^2 **RMSE**			🔓	🇺🇸	
Pei et al. [320]	**A** P R **F_1** AUC IR IRR MAE MAPE **MCC** MdAPE MRR MSE R^2 RMSE	🔓		🔓	🇺🇸	
Kumar et al. [330]	A P R F_1 AUC IR IRR MAE MAPE MCC MdAPE MRR **MSE** R^2 RMSE			🔓	🌍	
Hsu et al. [318]	**A** **P** R F_1 AUC IR IRR MAE MAPE MCC MdAPE **MRR** MSE R^2 RMSE			🔓	🇺🇸 🇹🇼	
Lazcano et al. [303]	A P R F_1 AUC IR IRR MAE **MAPE** MCC MdAPE MRR MSE **R^2** **RMSE**			🔒	🇺🇸	
Ang et al. [327]	A P R F_1 AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 **RMSE**		🔓	🔓	🇺🇸	
Sawhney et al. [342]	A P R F_1 AUC IR IRR MAE MAPE **MCC** MdAPE MRR **MSE** **R^2** RMSE		🔓		🇺🇸	
Liu et al. [324]	**A** **P** **R** **F_1** AUC IR IRR MAE MAPE **MCC** MdAPE MRR MSE R^2 RMSE	🔓	🔓		🇺🇸	
Zhao et al. [334]	**A** P R F_1 AUC IR IRR MAE MAPE MCC MdAPE MRR **MSE** R^2 **RMSE**		🔒	🔒	🇯🇵	
Ye et al. [344]	**A** **P** **R** **F_1** AUC IR IRR MAE MAPE **MCC** MdAPE MRR MSE R^2 RMSE	🔒		🔒	🇨🇳	
Liu et al. [38]	A P R F_1 AUC IR **IRR** MAE MAPE MCC MdAPE **MRR** **MSE** R^2 RMSE		🔓	🔓	🇺🇸	
Cheng et al. [306]	A P R **F_1** AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE	🔓	🔓	🔓	🇨🇳	✓
Cheng et al. [328]	**A** P R F_1 **AUC** IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE		🔓	🔒	🇺🇸	
Sawhney et al. [331]	**A** P R **F_1** AUC IR IRR MAE MAPE **MCC** MdAPE MRR MSE R^2 RMSE		🔓		🇺🇸	
Li et al. [335]	**A** P R F_1 AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE		🔒		🇯🇵	
Hu et al. [326]	**A** P R F_1 AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE	-	-		🇨🇳	
Sawhney et al. [297]	A P R F_1 AUC IR **IRR** MAE MAPE MCC MdAPE MRR MSE R^2 RMSE		🔓		🇺🇸 🇨🇳	
Li et al. [329]	**A** P R F_1 AUC IR IRR MAE MAPE **MCC** MdAPE MRR MSE R^2 RMSE		🔓		🇺🇸	
Deng et al. [36]	**A** P R **F_1** AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE		🔓	🔓	🇺🇸	
Qian et al. [298]	A **P** R F_1 AUC **IR** IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE	-		-	🇨🇳	
Shi et al. [340]	**A** P R **F_1** AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE	🔓		🔓	🇨🇳	
Li et al. [343]	**A** **P** **R** **F_1** AUC IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE	🔓		🔓	🇺🇸	✓
Zhao et al. [299]	**A** P R F_1 **AUC** IR IRR MAE MAPE MCC MdAPE MRR MSE R^2 RMSE	🔓	🔓		🇨🇳	
Hou et al. [336]	A **P** **R** **F_1** AUC IR IRR MAE **MAPE** MCC **MdAPE** MRR MSE R^2 RMSE			-	🇺🇸	
Xu et al. [332]	A P R F_1 AUC IR IRR **MAE** MAPE MCC **MdAPE** MRR MSE R^2 **RMSE**	🔓		🔓	🇨🇳	✓
Feng et al. [333]	A P R F_1 AUC IR IRR MAE MAPE MCC MdAPE **MRR** **MSE** R^2 RMSE	-		-	🇨🇳	
Gao et al. [321]	A P R F_1 AUC IR **IRR** MAE MAPE MCC MdAPE **MRR** **MSE** R^2 RMSE	🔓	🔓	🔓	🇺🇸	

major benchmarks such as the S&P 500, widely used as a proxy for large-cap U.S. equities; models trained on this dataset (e.g. [297], [343], [331]) benefit from high liquidity, abundant textual data, and structured reporting. Others target Asian markets, notably the Chinese CSI 100 and CSI 300 indices (e.g. [290], [326], [299]), which differ from Western markets in investor composition, regulation, transparency, and government influence—factors that shape the relative value of data modalities. Some studies extend to cross-market or multi-region scenarios, which pose challenges such as unequal data availability, linguistic variation, and mismatched trading calendars. A model pre-trained on U.S. data, for example, may underperform in emerging markets without domain adaptation. These findings underscore the need for context-aware modelling: a modality or architecture effective in one market may not generalise to others, making market choice (see *Market* column) central to multimodal forecasting research.

14.5 Open Challenges and Future Directions

Building on the insights of our taxonomy-driven review, we delineate persistent challenges in multimodal financial forecasting and outline promising research trajectories to address them.

Robustness to missing and misaligned modalities. In operational environments, data streams are frequently incomplete, delayed, or out of sync. Developing architectures that degrade gracefully under such conditions remains a primary objective. Future work should target principled handling of modality dropout and temporal misalignment, including uncertainty-aware training objectives and inference-time imputations that preserve calibration and reliability.

Standardised and adaptive fusion. Fusion practices range from naive concatenation to sophisticated attention mechanisms, yet the absence of shared evaluation protocols limits comparability. Research should pursue adaptive fusion schemes that weight modalities on the basis of data quality, recency, and task relevance, coupled with community benchmarks that enable fair, reproducible assessment across methods, markets, and horizons.

Integrating LLMs within multimodal pipelines. Large Language Models can enrich textual channels by extracting nuanced signals from news, social media, and financial filings, and by generating faithful natural-language rationales for forecasts. A productive direction is the tight coupling of LLMs with numerical, temporal, and graph-structured inputs, using interfaces that constrain generation to be decision-useful, auditable, and consistent with downstream quantitative components.

Interpretability and decision support. The composite nature of multimodal systems complicates post-hoc analysis. There is a pressing need for tools that attribute predictions to specific inputs and interactions across modalities and time. Such capabilities support risk management, model governance, and practitioner trust, and should be designed to operate under distribution shift and data sparsity.

Domain adaptation and transfer. Financial regimes vary across geographies, asset classes, and historical periods. Methods that adapt representations and decision rules to new domains, while controlling for drift and preserving performance guarantees, are essential for generalisability in heterogeneous and evolving settings.

Broader and higher-quality modalities. Augmenting models with well-curated alternative data—macroeconomic indicators, ESG signals, supply-chain events, and advanced sentiment measures—can expand the informational frontier. The emphasis should be on measurable incremental value, careful alignment and de-noising, and transparency about provenance and sampling biases.

Benchmarks and datasets at scale. Progress depends on accessible, large-scale benchmarks that reflect realistic constraints. Community resources should span multiple markets, asset classes, and modalities, include well-defined tasks and splits, and report costs as well as accuracy. Such artefacts will enable rigorous comparisons, ablations, and stress tests that assess robustness, scalability, and reproducibility.

Collectively, these directions chart a path toward multimodal systems that are more reliable under real-world constraints, more transparent in their reasoning, and more transferable across regimes and use cases.

14.6 Conclusion

This survey has presented a structured and comprehensive analysis of multimodal approaches for financial forecasting, covering developments from 2018 to 2025. By proposing a unified taxonomy along four dimensions—input modalities, modeling architectures, fusion strategies, and predictive tasks—we provided a systematic framework to compare 35 representative works. Our review highlights that integrating heterogeneous data sources such as market signals, textual news, and relational graphs can substantially enhance predictive performance, but also introduces new challenges, including temporal misalignment, modality imbalance, missing or noisy data, and limited cross-market generalization.

We also discussed interpretability as a critical dimension for practical deployment, particularly in the financial domain where transparency, accountability, and regulatory compliance are essential. Attention-based mechanisms and graph-based explanations have emerged as promising tools to uncover the contribution of different modalities and their interactions over time.

Looking forward, research in multimodal financial forecasting is likely to benefit from advances in adaptive fusion techniques, learning with incomplete modalities, and the integration of large language models and temporal graph neural networks. Furthermore, developing standardized benchmarks, robust evaluation protocols, and explainability frameworks will be essential to foster reproducibility and accelerate progress in this rapidly evolving field. By bridging methodological innovation with domain-specific constraints, future multimodal systems have the potential to deliver not only higher predictive accuracy but also greater interpretability and trustworthiness, ultimately supporting more informed and responsible financial decision-making.

Chapter 15

Conclusion

15.1 Summary of Contributions

This thesis has demonstrated the significant potential of domain-adapted Vector Space Models in addressing complex challenges in both Labour Market Intelligence and Financial Natural Language Processing. The research has established that specialized approaches, tailored to the unique characteristics of each domain, consistently outperform general-purpose language models while providing practical, scalable solutions for real-world applications.

Advances in Labour Market Intelligence The contributions in LMI have addressed critical gaps in European labour market analysis through three major innovations. The KRAKEN framework has proven highly effective for unsupervised skill identification as they appear in the OJAs, demonstrating strong performance on standard benchmarks with F_1-scores reaching 28.6% when considering the top-10 keyphrases. The approach successfully identified emerging skills across multiple European languages with accuracy rates between 56.8% and 76.4% based on expert evaluations. The JobSet dataset has provided a substantial resource for overcoming data scarcity, comprising over 15,000 synthetic job advertisements with significantly improved quality metrics, including perplexity and skill explicitness. The VEUCTOR framework has established a comprehensive methodology for cross-country model alignment, facilitating the generation and evaluation of thousands of embedding models across 28 European countries and enabling semantically consistent cross-national labour market analysis.

Advances in Financial NLP In the financial domain, this research has provided novel approaches for understanding modern market dynamics shaped by social trading platforms. The introduction of the Social Trading Action Detection (STAD) task and the accompanying FINREDDIT-2K benchmark dataset—containing 2,123 expert-annotated Reddit posts—has established a rigorous foundation for evaluating trading intent classification. Extensive benchmarking across 57 models revealed that several specialized architectures achieve F_1-scores exceeding 84%, with the best-performing model reaching 86.0% accuracy. The novel Content-based Centrality (CbC) metrics for assessing user reliability in financial social networks have demonstrated superior performance compared to traditional measures, achieving 60% alignment with human expert assessments. The systematic analysis of multimodal forecasting approaches has identified integration strategies that deliver substantial performance gains.

15.2 Limitations and Open Challenges

Despite the breadth of contributions presented in this thesis, several limitations must be acknowledged. First, many of the proposed approaches rely on large volumes of unstructured textual data,

such as online job advertisements and social media posts, which are inherently noisy, biased, and subject to platform-specific dynamics. While extensive preprocessing and evaluation strategies were adopted, the quality and representativeness of the input data remain a critical factor influencing model performance.

Second, although the thesis addresses multilingual and cross-country settings—particularly within the European labour market—the scope of empirical validation is still constrained to specific languages, platforms, and time periods. This limits the immediate generalizability of the findings to other geographical contexts or rapidly evolving market conditions. Similarly, several experiments are conducted in offline or retrospective settings, which do not fully capture the feedback loops and real-time dynamics present in operational financial and labour market systems.

Third, while the thesis advances embedding-based and multimodal perspectives, most models still rely on relatively static representations. Capturing long-term temporal dependencies, regime shifts, and abrupt market changes remains an open challenge. Moreover, the increasing complexity of multimodal architectures raises concerns related to interpretability, scalability, and computational cost, particularly in large-scale or real-time deployments.

Finally, influence modeling and sentiment-driven analysis in social finance are vulnerable to manipulation, coordinated behavior, and emerging forms of misinformation. Addressing robustness and trustworthiness in such environments remains an open research problem.

15.3 Future Research Directions

The limitations identified above naturally point to several promising directions for future research. A first avenue concerns the development of dynamic and temporally-aware models capable of adapting to evolving semantic patterns, market regimes, and social behaviors. Integrating temporal graph neural networks and sequential multimodal fusion strategies could significantly enhance predictive robustness.

A second direction involves advancing multimodal learning beyond modular pipelines toward fully end-to-end architectures that jointly model numerical, textual, and relational data. In this context, the integration of large language models and multimodal foundation models represents a particularly promising opportunity, both for representation learning and for improving interpretability through natural language explanations.

Future work should also focus on improving cross-domain and cross-market generalization, enabling models trained in one context to transfer knowledge effectively to others. Human-in-the-loop approaches, combining expert feedback with automated learning, could further enhance model reliability and practical adoption.

Finally, moving from offline evaluation to real-time and deployment-oriented settings constitutes a crucial step toward operational impact. This includes addressing scalability, latency constraints, and ethical considerations related to data usage and algorithmic decision-making. By pursuing these directions, future research can build upon the foundations laid by this thesis to develop more robust, adaptive, and trustworthy systems for labour market intelligence and financial analysis.

15.4 Concluding Remarks

This thesis has charted a path toward more effective, domain-aware natural language processing in two critical economic domains. By developing specialized methodologies that respect the unique characteristics of financial markets and labour ecosystems, while maintaining rigorous computational foundations, this research has demonstrated that targeted AI applications can provide substantial value in complex, real-world environments. The consistent empirical results across both domains underscore the importance of domain adaptation and specialized architectural choices.

The released datasets, frameworks, and empirical insights provide a foundation for continued advancement in both research and practice, contributing to the development of more responsive, data-driven economic systems. The work highlights the rich interdisciplinary ground that exists at the intersection of computational linguistics, labour economics, and quantitative finance. As both fields continue to evolve rapidly, the approaches developed here offer scalable, adaptable foundations for future innovation in understanding and navigating complex economic systems through advanced language technologies.

Bibliography

[1] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[2] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[3] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186.

[4] Mikhail Vinel et al. “Experimental Comparison of Unsupervised Approaches in the Task of Separating Specializations Within Professions in Job Vacancies”. In: *Conference on Artificial Intelligence and Natural Language*. Springer. 2019, pp. 99–112.

[5] Anna Giabelli et al. “NEO: A System for Identifying New Emerging Occupation from Job Ads”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 18. 2021, pp. 16035–16037.

[6] Simone D’Amico et al. “Enriching Skill Taxonomies through Vector Space Models”. In: *2024 IEEE International Conference on Big Data (BigData)*. IEEE. 2024, pp. 2297–2302.

[7] EuroStat. *Towards the European Web Intelligence Hub — European System for Collection and Analysis of Online Job Advertisement Data (WIH-OJA), available at* `https://tinyurl.com/y3xqzfhp`. 2020.

[8] Roberto Boselli et al. “WoLMIS: a labor market intelligence system for classifying web job vacancies”. In: *J. Intell. Inf. Syst.* 51.3 (2018), pp. 477–502.

[9] Roberto Boselli et al. “Using Machine Learning for Labour Market Intelligence”. In: *ECML PKDD 2017: Machine Learning and Knowledge Discovery in Database* (2017), pp. 330–342.

[10] Emilio Colombo, Fabio Mercorio, and Mario Mezzanzanica. “AI meets labor market: Exploring the link between automation and skills”. In: *Information Economics and Policy* 47 (2019), pp. 27–37.

[11] Roberto Boselli et al. “Classifying online job advertisements through machine learning”. In: *Future Generation Computer Systems* 86 (2018), pp. 319–328.

[12] Francesco Colace et al. “Towards Labour Market Intelligence through Topic Modelling”. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS)*. 2019, pp. 5256–5265. ISBN: 978-0-9981331-2-6. URL: `http://hdl.handle.net/10125/59962`.

[13] Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. “A survey on sentiment analysis methods, applications, and challenges”. In: *Artificial Intelligence Review* 55.7 (2022), pp. 5731–5780. DOI: `https://doi.org/10.1007/s10462-022-10144-1`.

[14] Ankit Thakkar and Kinjal Chaudhari. "Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions". In: *Information Fusion* 65 (2021), pp. 95–107. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2020.08.019.

[15] Qian Li et al. "A survey on text classification: From shallow to deep learning". In: *arXiv preprint arXiv:2008.00364* (2020).

[16] Antonino Ferraro and Giancarlo Sperlì. "How does user-generated content on Social Media affect stock predictions? A case study on GameStop". In: *Online Social Networks and Media* 43-44 (2024), p. 100293. ISSN: 2468-6964. DOI: https://doi.org/10.1016/j.osnem.2024.100293.

[17] Zellig S Harris. *Methods in structural linguistics.* University of Chicago Press, 1951.

[18] Zellig S Harris. "Distributional structure". In: *Word* 10.2-3 (1954), pp. 146–162.

[19] Alessandro Lenci and Magnus Sahlgren. *Distributional semantics*. Cambridge University Press, 2023.

[20] Roberto Boselli, Simone D'Amico, and Navid Nobani. "eXplainable AI for word embeddings: A survey". In: *Cognitive Computation* 17.1 (2025), p. 19.

[21] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26 (2013).

[22] Piotr Bojanowski et al. "Enriching word vectors with subword information". In: *Transactions of the association for computational linguistics* 5 (2017), pp. 135–146.

[23] Matthew E. Peters et al. "Deep Contextualized Word Representations". In: *ArXiv* abs/1802.05365 (2018). URL: https://api.semanticscholar.org/CorpusID:3626819.

[24] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[25] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[26] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[27] Abhimanyu Dubey et al. "The llama 3 herd of models". In: *arXiv e-prints* (2024), arXiv–2407.

[28] Hugo Touvron et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

[29] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[30] Tim Loughran and Bill McDonald. "Measuring readability in financial disclosures". In: *the Journal of Finance* 69.4 (2014), pp. 1643–1671.

[31] Junwen Duan et al. "Learning target-specific representations of financial news documents for cumulative abnormal return prediction". In: *Proceedings of the 27th international conference on computational linguistics*. 2018, pp. 2823–2833.

[32] Di Luo et al. "Causality-guided multi-memory interaction network for multivariate stock price movement prediction". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 12164–12176.

[33] Yumo Xu and Shay B Cohen. "Stock movement prediction from tweets and historical prices". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 1970–1979.

[34] Yu Qin and Yi Yang. "What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 390–401.

[35] Dogu Araci. "Finbert: Financial sentiment analysis with pre-trained language models". In: *arXiv preprint arXiv:1908.10063* (2019).

[36] Shumin Deng et al. "Knowledge-driven stock trend prediction and explanation via temporal convolutional network". In: *Companion proceedings of the 2019 world wide web conference*. 2019, pp. 678–685.

[37] Xiao Ding et al. "Deep learning for event-driven stock prediction." In: *Ijcai*. Vol. 15. 2015, pp. 2327–2333.

[38] Ying Liu et al. "Multi-Modal Temporal Dynamic Graph Construction for Stock Rank Prediction". In: *Mathematics* 13.5 (2025), p. 845.

[39] Arthur Turrell et al. "Using job vacancies to understand the effects of labour market mismatch on UK output and productivity". In: (2018).

[40] Maria Papoutsoglou et al. "Extracting Knowledge from on-line Sources for Software Engineering Labor Market: A Mapping Study". In: *IEEE Access* (2019).

[41] Faizan Javed et al. "Large-scale occupational skills normalization for online recruitment". In: *Twenty-Ninth IAAI Conference*. 2017.

[42] UK Commission for Employment and Skills. *The Importance of LMI, available at* `https://goo.gl/TtRwvS`. 2015.

[43] Anna Giabelli et al. "Skills2Job: A recommender system that encodes job offer embeddings on graph databases". In: *Applied Soft Computing* 101 (2021), p. 107049.

[44] José Azar et al. "Concentration in US labor markets: Evidence from online vacancy data". In: *Labour Economics* 66 (2020), p. 101886. ISSN: 0927-5371. DOI: https://doi.org/10.1016/j.labeco.2020.101886.

[45] José Azar et al. "Minimum Wage Employment Effects and Labor Market Concentration". In: *The Review of Economic Studies* (Sept. 2023), rdad091. ISSN: 0034-6527. DOI: 10.1093/restud/rdad091. (Visited on 09/14/2023).

[46] Brad Hershbein and Lisa B. Kahn. "Do Recessions Accelerate Routine-Biased Technological Change? Evidence from Vacancy Postings". In: *American Economic Review* 108.7 (July 2018), pp. 1737–72. DOI: 10.1257/aer.20161570.

[47] Emilio Colombo and Alberto Marcato. "Skill demand and labour market concentration: evidence from Italian vacancies". In: *International Journal of Manpower* 44.9 (Oct. 2023), pp. 156–198. DOI: 10.1108/IJM-04-2023-0181.

[48] J. Carter Braxton and Bledi Taska. "Technological Change and the Consequences of Job Loss". In: *American Economic Review* 113.2 (Feb. 2023), pp. 279–316. DOI: 10.1257/aer.20210182. URL: https://www.aeaweb.org/articles?id=10.1257/aer.20210182.

[49] Ran Gu and Ling Zhong. "Effects of stay-at-home orders on skill requirements in vacancy postings". In: *Labour Economics* 82 (2023), p. 102342. ISSN: 0927-5371. DOI: https://doi.org/10.1016/j.labeco.2023.102342.

[50] Avi Goldfarb, Bledi Taska, and Florenta Teodoridis. "Could machine learning be a general purpose technology? A comparison of emerging technologies using data from online job postings". In: *Research Policy* 52.1 (2023), p. 104653. ISSN: 0048-7333. DOI: https://doi.org/10.1016/j.respol.2022.104653.

[51] Alicia Sasser Modestino, Daniel Shoag, and Joshua Ballance. "Upskilling: Do Employers Demand Greater Skill When Workers Are Plentiful?" In: *The Review of Economics and Statistics* 102.4 (Oct. 2020), pp. 793–805. DOI: 10.1162/rest_a_00835.

[52] OECD (2017). *In-Depth Analysis of the Labour Market Relevance and Outcomes of Higher Education Systems: Analytical Framework and Country Practices Report*. Enhancing Higher Education System Performance, OECD, Paris.

[53] CEDEFOP. *Real-time Labour Market information on Skill Requirements: Setting up the EU system for online vacancy analysis.* https://goo.gl/5FZS3E. 2016.

[54] H Charles J Godfray. "Challenges for taxonomy". In: *Nature* 417.6884 (2002), pp. 17–19.

[55] Anna Giabelli et al. "NEO: A Tool for Taxonomy Enrichment with New Emerging Occupations". In: *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*. Ed. by Jeff Z. Pan et al. Vol. 12507. Lecture Notes in Computer Science. Springer, 2020, pp. 568–584. DOI: 10.1007/978-3-030-62466-8_35. URL: https://doi.org/10.1007/978-3-030-62466-8%5C_35.

[56] Anna Giabelli et al. "WETA: Automatic taxonomy alignment via word embeddings". In: *Computers in Industry* 138 (2022), p. 103626.

[57] Alexander Maedche and Steffen Staab. "Ontology learning for the semantic web". In: *IEEE Intelligent systems* 16.2 (2001), pp. 72–79.

[58] Social Affairs Directorate-General for Employment and Inclusion (European Commission). *ESCO handbook*. English. Version Paris. 2019.

[59] Claudia Leacock, Martin Chodorow, and George A Miller. "Using corpus statistics and WordNet relations for sense identification". In: *Computational Linguistics* 24.1 (1998), pp. 147–165.

[60] Jay J Jiang and David W Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy". In: *arXiv preprint cmp-lg/9709008* (1997).

[61] Nuno Seco, Tony Veale, and Jer Hayes. "An intrinsic information content metric for semantic similarity in WordNet". In: *Ecai*. Vol. 16. 2004, p. 1089.

[62] Dekang Lin et al. "An information-theoretic definition of similarity." In: *Icml*. Vol. 98. 1998. 1998, pp. 296–304.

[63] Lorenzo Malandri et al. "Taxoref: Embeddings evaluation for ai-driven taxonomy refinement". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2021, pp. 612–627.

[64] Philip Resnik. "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language". In: *Journal of artificial intelligence research* 11 (1999), pp. 95–130.

[65] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160.

[66] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information fusion* 58 (2020), pp. 82–115.

[67] Nadia Burkart and Marco F Huber. "A survey on the explainability of supervised machine learning". In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317.

[68] Gesina Schwalbe and Bettina Finzel. "A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts". In: *Data Mining and Knowledge Discovery* (2023), pp. 1–59.

[69] Weiping Ding et al. “Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey”. In: *Information Sciences* 615 (2022), pp. 238–292.

[70] Rudresh Dwivedi et al. “Explainable AI (XAI): Core ideas, techniques, and solutions”. In: *ACM Computing Surveys* 55.9 (2023), pp. 1–33.

[71] Waddah Saeed and Christian Omlin. “Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities”. In: *Knowledge-Based Systems* 263 (2023), p. 110273.

[72] Frank van Mourik et al. “Tertiary Review on Explainable Artificial Intelligence: Where Do We Stand?” In: *Machine Learning and Knowledge Extraction* 6.3 (2024), pp. 1997–2017.

[73] George EP Box et al. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[74] Eugene F Fama. “Efficient capital markets: A review of theory and empirical work”. In: *The journal of Finance* 25.2 (1970), pp. 383–417.

[75] Paul C Tetlock. “Giving content to investor sentiment: The role of media in the stock market”. In: *The Journal of finance* 62.3 (2007), pp. 1139–1168.

[76] Robert J Shiller et al. “Do stock prices move too much to be justified by subsequent changes in dividends?” In: (1981).

[77] Hongren Wang et al. “Multimodal market information fusion for stock price trend prediction in the pharmaceutical sector”. In: *Applied Intelligence* 55.1 (2025), p. 77.

[78] Chang Zong et al. “Stock movement prediction with multimodal stable fusion via gated cross-attention mechanism”. In: *Complex & Intelligent Systems* 11.9 (2025), p. 396.

[79] Yankai Sheng, Yuanyu Qu, and Ding Ma. “Stock price crash prediction based on multimodal data machine learning models”. In: *Finance Research Letters* 62 (2024), p. 105195.

[80] Yang Zhang et al. “Camef: Causal-augmented multi-modality event-driven financial forecasting by integrating time series patterns and salient macroeconomic announcements”. In: *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 2025, pp. 3867–3878.

[81] Samyak Jain et al. “Saliency-Aware Interpolative Augmentation for Multimodal Financial Prediction”. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024, pp. 14285–14297.

[82] Allen H Huang, Hui Wang, and Yi Yang. “FinBERT: A large language model for extracting information from financial text”. In: *Contemporary Accounting Research* 40.2 (2023), pp. 806–841.

[83] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. “FinGPT: Open-Source Financial Large Language Models”. In: *FinLLM Symposium at IJCAI 2023* (2023).

[84] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[85] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[86] Fuli Feng et al. “Temporal relational ranking for stock prediction”. In: *ACM Transactions on Information Systems (TOIS)* 37.2 (2019), pp. 1–30.

[87] Kisu Lee et al. “Hierarchical multi-modal fusion architecture search for stock market forecasting”. In: *Applied Soft Computing* (2025), p. 113581.

[88] Fatima Dakalbab et al. “Advancing forex prediction through multimodal text-driven model and attention mechanisms”. In: *Intelligent Systems with Applications* 26 (2025), p. 200518.

[89] Manali Patel, Krupa Jariwala, and Chiranjoy Chattopadhyay. "A Systematic Review on Graph Neural Network-based Methods for Stock Market Forecasting". In: *ACM Computing Surveys* 57.2 (2024), pp. 1–38.

[90] Jianian Wang et al. "A Review on Graph Neural Network Methods in Financial Applications". In: *Journal of Data Science* 20.2 (2022), pp. 111–134.

[91] Yushan Jiang et al. "Multi-modal time series analysis: A tutorial and survey". In: *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 2025, pp. 6043–6053.

[92] Yajiao Tang et al. "A survey on machine learning models for financial time series forecasting". In: *Neurocomputing* 512 (2022), pp. 363–380. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2022.09.003.

[93] Noella Nazareth and Yeruva Venkata Ramana Reddy. "Financial applications of machine learning: A literature review". In: *Expert Systems with Applications* 219 (2023), p. 119640. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.119640.

[94] Wenbo Ge et al. "Neural Network–Based Financial Volatility Forecasting: A Systematic Review". In: *ACM Comput. Surv.* 55.1 (Jan. 2022). ISSN: 0360-0300. DOI: 10.1145/3483596.

[95] Ankit Thakkar and Kinjal Chaudhari. "A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions". In: *Expert Systems with Applications* 177 (2021), p. 114800. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.114800.

[96] Kenniy Olorunnimbe and Herna Viktor. "Deep learning in the stock market—a systematic survey of practice, backtesting, and applications". In: *Artificial Intelligence Review* 56.3 (2023), pp. 2057–2109. DOI: https://doi.org/10.1007/s10462-022-10226-0.

[97] Min Choi et al. "Stock price momentum modeling using social media data". In: *Expert Systems with Applications* 237 (2024), p. 121589. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.121589.

[98] Xiang Ma et al. "Fuzzy hypergraph network for recommending top-K profitable stocks". In: *Information Sciences* 613 (2022), pp. 239–255. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.09.010.

[99] Wei-Chia Huang et al. "Attentive gated graph sequence neural network-based time-series information fusion for financial trading". In: *Information Fusion* 91 (2023), pp. 261–276. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2022.10.006.

[100] Mattia Birti, Francesco Osborne, and Andrea Maurino. "Optimizing Large Language Models for ESG Activity Detection in Financial Texts". In: *arXiv preprint arXiv:2502.21112* (2025).

[101] Surupendu Gangopadhyay and Prasenjit Majumder. "Text representation for direction prediction of share market". In: *Expert Systems with Applications* 211 (2023), p. 118472. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.118472.

[102] Peder Gjerstad et al. "Do President Trump's tweets affect financial markets?" In: *Decision Support Systems* 147 (2021), p. 113577. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2021.113577.

[103] Hans Christian Schmitz et al. "When machines trade on corporate disclosures: Using text analytics for investment strategies". In: *Decision Support Systems* 165 (2023), p. 113892. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2022.113892.

[104] Qiuyue Zhang et al. "Incorporating stock prices and text for stock movement prediction based on information fusion". In: *Engineering Applications of Artificial Intelligence* 127 (2024), p. 107377. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2023.107377.

[105] Hang Dong et al. “How are social and mass media different in relation to the stock market? A study on topic coverage and predictive value”. In: *Information & Management* 59.2 (2022), p. 103588. ISSN: 0378-7206. DOI: https://doi.org/10.1016/j.im.2021.103588.

[106] Xiangyu Li et al. “FinReport: Explainable Stock Earnings Forecasting via News Factor Analyzing Model”. In: *Companion Proceedings of the ACM Web Conference 2024*. WWW ’24. Singapore, Singapore: Association for Computing Machinery, 2024, pp. 319–327. ISBN: 9798400701726. DOI: 10.1145/3589335.3648330. URL: https://doi.org/10.1145/3589335.3648330.

[107] Qianqian Xie et al. “FinBen: A Holistic Financial Benchmark for Large Language Models”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 95716–95743.

[108] Kelvin Du et al. “Financial Sentiment Analysis: Techniques and Applications”. In: *ACM Comput. Surv.* 56.9 (Apr. 2024). ISSN: 0360-0300. DOI: 10.1145/3649451.

[109] Chuan Qin et al. “FollowAKOInvestor: Stock recommendation by hearing voices from all kinds of investors with machine learning”. In: *Expert Systems with Applications* 249 (2024), p. 123522. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2024.123522.

[110] Yong Zhuang et al. “Leveraging large language models to examine the interaction between investor sentiment and stock performance”. In: *Engineering Applications of Artificial Intelligence* 150 (2025), p. 110602. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2025.110602.

[111] Qianqian Xie et al. “PIXIU: a large language model, instruction data and evaluation benchmark for finance”. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS ’23. New Orleans, LA, USA: Curran Associates Inc., 2024.

[112] Shijie Wu et al. “Bloomberggpt: A large language model for finance”. In: *arXiv preprint arXiv:2303.17564* (2023).

[113] Zihan Dong, Xinyu Fan, and Zhiyuan Peng. “FNSPID: A Comprehensive Financial News Dataset in Time Series”. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD ’24. Barcelona, Spain: Association for Computing Machinery, 2024, pp. 4918–4927. ISBN: 9798400704901. DOI: 10.1145/3637528.3671629. URL: https://doi.org/10.1145/3637528.3671629.

[114] Gabriele Ranco et al. “The effects of Twitter sentiment on stock price returns”. In: *PloS one* 10.9 (2015), e0138441.

[115] J Anthony Cookson and Marina Niessner. “Why don’t we agree? Evidence from a social network of investors”. In: *The Journal of Finance* 75.1 (2020), pp. 173–228.

[116] Danqi Hu et al. “The rise of reddit: How social media affects retail investors and short-sellers’ roles in price discovery”. In: *Available at SSRN 3807655* (2021).

[117] Štefan Lyócsa, Eduard Baumöhl, and Tomáš Vỳrost. “YOLO trading: Riding with the herd during the GameStop episode”. In: *Finance Research Letters* 46 (2022), p. 102359.

[118] Hailiang Chen et al. “Wisdom of crowds: The value of stock opinions transmitted through social media”. In: *The review of financial studies* 27.5 (2014), pp. 1367–1403.

[119] Suwan Long et al. ““I just like the stock”: The role of Reddit sentiment in the GameStop share rally”. In: *Financial Review* 58.1 (2023), pp. 19–37.

[120] Maxime LD Nicolas. “Estimating a model of herding behavior on social networks”. In: *Physica A: Statistical Mechanics and its Applications* 604 (2022), p. 127884.

[121] Guocheng Wang and Yanyi Wang. “Herding, social network and volatility”. In: *Economic Modelling* 68 (2018), pp. 74–81.

[122] Chee Wei Phang, Chenghong Zhang, and Juliana Sutanto. "The influence of user interaction and participation in social media on the consumption intention of niche products". In: *Information & Management* 50.8 (2013), pp. 661–672.

[123] Cody Buntain and Jennifer Golbeck. "Identifying social roles in reddit using network structure". In: *Proceedings of the 23rd international conference on world wide web*. 2014, pp. 615–620.

[124] Stephen P Borgatti. "Centrality and network flow". In: *Social networks* 27.1 (2005), pp. 55–71.

[125] Daifeng Li et al. "Analyzing stock market trends using social media user moods and social influence". In: *Journal of the Association for Information Science and Technology* 70.9 (2019), pp. 1000–1013.

[126] Arnav Machavarapu. "Reddit sentiments effects on stock market prices". In: *Smart Intelligent Computing and Applications, Volume 1: Proceedings of Fifth International Conference on Smart Computing and Informatics (SCI 2021)*. Springer. 2022, pp. 75–84.

[127] Yousra Fettach, Adil Bahaj, and Mounir Ghogho. "JobEdKG: An uncertain knowledge graph-based approach for recommending online courses and predicting in-demand skills based on career choices". In: *EAAI* 131 (2024), p. 107779.

[128] Lorenzo Malandri et al. "MEET-LM: A method for embeddings evaluation for taxonomic data in the labour market". In: *Computers in Industry* 124 (2021), p. 103341.

[129] Jens-Joris Decorte et al. "Jobbert: Understanding job titles through skills". In: *arXiv preprint arXiv:2109.09605* (2021).

[130] Mike Zhang, Rob Van Der Goot, and Barbara Plank. "ESCOXLM-R: Multilingual taxonomy-driven pre-training for the job market domain". In: *arXiv preprint arXiv:2305.12092* (2023).

[131] Akshay Bhola et al. "Retrieving skills from job descriptions: A language model based extreme multi-label classification framework". In: *Proceedings of the 28th international conference on computational linguistics*. 2020, pp. 5832–5842.

[132] Jens-Joris Decorte et al. "Extreme multi-label skill extraction training using large language models". In: *arXiv preprint arXiv:2307.10778* (2023).

[133] Benjamin Clavié and Guillaume Soulié. "Large language models as batteries-included zero-shot esco skills matchers". In: *arXiv preprint arXiv:2307.03539* (2023).

[134] Ziwei Ji et al. "Survey of hallucination in natural language generation". In: *ACM Computing Surveys* 55.12 (2023), pp. 1–38.

[135] Wenhao Yu et al. "A survey of knowledge-enhanced text generation". In: *ACM Computing Surveys* 54.11s (2022), pp. 1–38.

[136] Nayeon Lee et al. "Factuality enhanced language models for open-ended text generation". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 34586–34599.

[137] Ariana Martino, Michael Iannelli, and Coleen Truong. "Knowledge injection to counter large language model (LLM) hallucination". In: *European Semantic Web Conference*. Springer. 2023, pp. 182–185.

[138] Shirui Pan et al. "Unifying large language models and knowledge graphs: A roadmap". In: *IEEE Transactions on Knowledge and Data Engineering* (2024).

[139] Veniamin Veselovsky et al. "Generating Faithful Synthetic Data with Large Language Models: A Case Study in Computational Social Science". In: *arXiv preprint arXiv:2305.15041* (2023).

[140] John Chung, Ece Kamar, and Saleema Amershi. “Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 575–593.

[141] Yue Yu et al. “Large language model as attributed training data generator: A tale of diversity and bias”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[142] Luiza Sayfullina, Eric Malmi, and Juho Kannala. “Learning representations for soft skill matching”. In: *Analysis of Images, Social Networks and Texts: 7th International Conference, AIST 2018, Moscow, Russia, July 5–7, 2018, Revised Selected Papers 7*. Springer. 2018, pp. 141–152.

[143] Ellery Smith et al. “Syntax-based skill extractor for job advertisements”. In: *2019 6th Swiss Conference on Data Science (SDS)*. IEEE. 2019, pp. 80–81.

[144] Akshay Gugnani and Hemant Misra. “Implicit skills extraction using document embedding and its use in job recommendation”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 08. 2020, pp. 13286–13293.

[145] Damian A Tamburri, Willem-Jan Van Den Heuvel, and Martin Garriga. “Dataops for societal intelligence: a data pipeline for labor market skills extraction and matching”. In: *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE. 2020, pp. 391–394.

[146] Mike Zhang et al. “SkillSpan: Hard and Soft Skill Extraction from English Job Postings”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022, pp. 4962–4984.

[147] Antoine Magron et al. “JobSkape: A Framework for Generating Synthetic Job Postings to Enhance Skill Matching”. In: *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*. 2024, pp. 43–58.

[148] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. “A survey of cross-lingual word embedding models”. In: *Journal of Artificial Intelligence Research* 65 (2019), pp. 569–631.

[149] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. “Exploiting similarities among languages for machine translation”. In: *arXiv* (2013).

[150] Zewen Chi et al. “Improving Pretrained Cross-Lingual Language Models via Self-Labeled Word Alignment”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 3418–3430.

[151] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. “A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings”. In: *arXiv preprint arXiv:1805.06297* (2018).

[152] Ion Madrazo Azpiazu and Maria Soledad Pera. “Hierarchical mapping for crosslingual word embedding alignment”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 361–376.

[153] Alexis Conneau et al. “Word translation without parallel data”. In: *arXiv preprint arXiv:1710.04087* (2017).

[154] Lorenzo Malandri et al. “SeNSe: embedding alignment via semantic anchors selection”. In: *International Journal of Data Science and Analytics* (2024), pp. 1–15.

[155] Julia El Zini and Mariette Awad. “On the explainability of natural language processing deep models”. In: *ACM Computing Surveys* 55.5 (2022), pp. 1–31.

[156] Yan Zheng et al. “Embeddingtree: Hierarchical exploration of entity features in embedding”. In: *2023 IEEE 16th Pacific Visualization Symposium (PacificVis)*. IEEE. 2023, pp. 217–221.

[157] Hongyin Luo et al. “Online learning of interpretable word embeddings”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1687–1692.

[158] Alona Fyshe et al. “A compositional and interpretable semantic space”. In: *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies*. 2015, pp. 32–41.

[159] Stephen Boyd et al. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine learning* 3.1 (2011), pp. 1–122.

[160] MFYTD Yogatama and CDNA Smith. “Sparse overcomplete word vector representations”. In: *ACL*. 2015.

[161] Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. “Ultradense Word Embeddings by Orthogonal Transformation”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, pp. 767–777.

[162] Sascha Rothe and Hinrich Schütze. “Word embedding calculus in meaningful ultradense subspaces”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2016, pp. 512–517.

[163] Martin Andrews. “Compressing word embeddings”. In: *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part IV 23*. Springer. 2016, pp. 413–422.

[164] Stuart Lloyd. “Least squares quantization in PCM”. In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[165] Kyoung-Rok Jang and Sung-Hyon Myaeng. “Elucidating conceptual properties from word embeddings”. In: *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*. 2017, pp. 91–95.

[166] Brian Murphy, Partha Talukdar, and Tom Mitchell. “Learning effective and interpretable semantic models using non-negative sparse embedding”. In: *Proceedings of COLING 2012*. 2012, pp. 1933–1950.

[167] Ivan Vulić et al. “Hyperlex: A large-scale evaluation of graded lexical entailment”. In: *Computational Linguistics* 43.4 (2017), pp. 781–835.

[168] Lütfi Kerem Şenel et al. “Semantic structure and interpretability of word embeddings”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.10 (2018), pp. 1769–1779.

[169] Anil Bhattacharyya. “On a measure of divergence between two statistical populations defined by their probability distribution”. In: *Bulletin of the Calcutta Mathematical Society* 35 (1943), pp. 99–110.

[170] Valentin Trifonov et al. “Learning and Evaluating Sparse Interpretable Sentence Embeddings”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 2018, pp. 200–210.

[171] Anant Subramanian et al. “Spine: Sparse interpretable neural embeddings”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[172] Carl Allen and Timothy Hospedales. “Analogies explained: Towards understanding word embeddings”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 223–231.

[173] Piero Molino, Yang Wang, and Jiawei Zhang. “Parallax: Visualizing and Understanding the Semantics of Embedding Spaces via Algebraic Formulae”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2019, pp. 165–180.

[174] Adly Templeton. “Word Equations: Inherently Interpretable Sparse Word Embeddings through Sparse Coding”. In: *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. 2021, pp. 177–191.

[175] Diego Garcia-Olano et al. “Intermediate Entity-based Sparse Interpretable Representation Learning”. In: *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. 2022, pp. 210–224.

[176] Lütfi Kerem Şenel et al. “Imparting interpretability to word embeddings while preserving semantic structure”. In: *Natural Language Engineering* 27.6 (2021), pp. 721–746.

[177] Binny Mathew et al. “The polar framework: Polar opposites enable interpretability of pre-trained word embeddings”. In: *Proceedings of the Web Conference 2020*. 2020, pp. 1548–1558.

[178] Peter Mark Roget. *Roget’s Thesaurus of English Words and Phrases...* TY Crowell Company, 1911.

[179] Lütfi Kerem Şenel et al. “Learning interpretable word embeddings via bidirectional alignment of dimensions with semantic concepts”. In: *Information Processing & Management* 59.3 (2022), p. 102925.

[180] Jan Engler et al. “SensePOLAR: Word sense aware interpretability for pre-trained contextual word embeddings”. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. 2022, pp. 4607–4619.

[181] M Atif Qureshi and Derek Greene. “EVE: explainable vector based embedding technique using Wikipedia”. In: *Journal of Intelligent Information Systems* 53 (2019), pp. 137–165.

[182] Manaal Faruqui et al. “Retrofitting Word Vectors to Semantic Lexicons”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2015.

[183] George A Miller. “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[184] Collin F Baker, Charles J Fillmore, and John B Lowe. “The berkeley framenet project”. In: *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*. 1998.

[185] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. “PPDB: The paraphrase database”. In: *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*. 2013, pp. 758–764.

[186] Jieyu Zhao et al. “Learning Gender-Neutral Word Embeddings”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2018.

[187] Alexander Panchenko. “Best of both worlds: Making word sense embeddings interpretable”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. 2016, pp. 2649–2655.

[188] Eric H Huang et al. “Improving word representations via global context and multiple word prototypes”. In: *Proceedings of the 50th annual meeting of the association for computational linguistics (Volume 1: Long papers)*. 2012, pp. 873–882.

[189] Roberto Navigli and Simone Paolo Ponzetto. “BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network”. In: *Artificial intelligence* 193 (2012), pp. 217–250.

[190] Sergey Bartunov et al. “Breaking sticks and ambiguities with adaptive skip-gram”. In: *artificial intelligence and statistics*. PMLR. 2016, pp. 130–138.

[191] Kishlay Jha et al. “Interpretable word embeddings for medical domain”. In: *2018 IEEE international conference on data mining (ICDM)*. IEEE. 2018, pp. 1061–1066.

[192] Adam Lauretig. “Identification, Interpretability, and Bayesian Word Embeddings”. In: *Proceedings of the Third Workshop on Natural Language Processing and Computational Social Science*. 2019, pp. 7–17.

[193] Miriam Hurtado Bodell, Martin Arvidsson, and Måns Magnusson. “Interpretable Word Embeddings via Informative Priors”. In: *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics. 2019, pp. 6324–6330.

[194] Jingyao Tang et al. “Sparse Variational Autoencoder-Based Interpretable Bimodal Word Embeddings”. In: *2021 International Conference on Machine Learning and Cybernetics (ICMLC)*. IEEE. 2021, pp. 1–6.

[195] Jie Li and Chun-qi Zhou. “Incorporation of human knowledge into data embeddings to improve pattern significance and interpretability”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.1 (2022), pp. 723–733.

[196] Sungjoon Park, JinYeong Bak, and Alice Oh. “Rotated word vector representations and their interpretability”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 401–411.

[197] Alexey Zobnin. “Rotations and interpretability of word embeddings: The case of the Russian language”. In: *Analysis of Images, Social Networks and Texts: 6th International Conference, AIST 2017, Moscow, Russia, July 27–29, 2017, Revised Selected Papers 6*. Springer. 2018, pp. 116–128.

[198] Philipp Dufter and Hinrich Schütze. “Analytical Methods for Interpretable Ultradense Word Embeddings”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 1185–1191.

[199] Kawin Ethayarajh. “Rotate King to get Queen: Word Relationships as Orthogonal Transformations in Embedding Space”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 3503–3508.

[200] James Foulds. “Mixed membership word embeddings for computational social science”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 86–95.

[201] Joseph Reisinger and Raymond Mooney. “Multi-prototype vector-space models of word meaning”. In: *Human Language Technologies: the 2010 annual conference of the north american chapter of the association for computational linguistics*. 2010, pp. 109–117.

[202] Maria Pelevina et al. “Making Sense of Word Embeddings”. In: *Proceedings of the 1st Workshop on Representation Learning for NLP*. 2016, pp. 174–183.

[203] Lauro Snidaro, Giovanni Ferrin, and Gian Luca Foresti. "Distributional memory explainable word embeddings in continuous space". In: *2019 22th International Conference on Information Fusion (FUSION)*. IEEE. 2019, pp. 1–7.

[204] Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. "Word2Sense: Sparse interpretable word embeddings". In: *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*. 2019, pp. 5692–5705.

[205] Jamin Shin, Andrea Madotto, and Pascale Fung. "Interpreting word embeddings with eigenvector analysis". In: *32nd Conference on Neural Information Processing Systems (NIPS 2018), IRASL workshop*. 2018, pp. 73–81.

[206] Haitong Zhang et al. "Improving interpretability of word embeddings by generating definition and usage". In: *Expert Systems with Applications* 160 (2020), p. 113633.

[207] Juri Opitz and Anette Frank. "SBERT studies Meaning Representations: Decomposing Sentence Embeddings into Explainable Semantic Features". In: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2022, pp. 625–638.

[208] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 3982–3992.

[209] Laura Banarescu et al. "Abstract meaning representation for sembanking". In: *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*. 2013, pp. 178–186.

[210] Yiyu Yao. "The art of granular computing". In: *Rough Sets and Intelligent Systems Paradigms: International Conference, RSEISP 2007, Warsaw, Poland, June 28-30, 2007. Proceedings 1*. Springer. 2007, pp. 101–112.

[211] Jing Tao Yao, Athanasios V Vasilakos, and Witold Pedrycz. "Granular computing: perspectives and challenges". In: *IEEE Transactions on Cybernetics* 43.6 (2013), pp. 1977–1989.

[212] Tsau-Young Lin. "Granular computing: practices, theories, and future directions". In: *Granular, Fuzzy, and Soft Computing*. Springer, 2023, pp. 199–219.

[213] Erik Cambria et al. "A survey on XAI and natural language explanations". In: *Information Processing & Management* 60.1 (2023), p. 103111.

[214] Simone D'Amico et al. "KRAKEN: A Novel Semantic-Based Approach for Keyphrases Extraction." In: *KDIR*. 2023, pp. 289–297.

[215] Anna Giabelli et al. "Skills2Graph: Processing million Job Ads to face the Job Skill Mismatch Problem." In: *IJCAI*. 2021, pp. 4984–4987.

[216] Shailendra Singh Kathait et al. "Unsupervised key-phrase extraction using noun phrases". In: *International Journal of Computer Applications* 162.1 (2017), pp. 1–5.

[217] Mario Mezzanzanica et al. "A model-based approach for developing data cleansing solutions". In: *Journal of Data and Information Quality (JDIQ)* 5.4 (2015), pp. 1–28.

[218] Mario Mezzanzanica et al. "Data Quality Sensitivity Analysis on Aggregate Indicators". In: *DATA 2012 - Proceedings of the International Conference on Data Technologies and Applications, Rome, Italy, 25-27 July, 2012*. Ed. by Markus Helfert, Chiara Francalanci, and Joaquim Filipe. SciTePress, 2012, pp. 97–108.

[219] Roberto Boselli et al. “A Policy-Based Cleansing and Integration Framework for Labour and Healthcare Data”. In: *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics - State-of-the-Art and Future Challenges*. Ed. by Andreas Holzinger and Igor Jurisica. Vol. 8401. Lecture Notes in Computer Science. Springer, 2014, pp. 141–168. DOI: 10.1007/978-3-662-43968-5_8. URL: https://doi.org/10.1007/978-3-662-43968-5%5C_8.

[220] Su Nam Kim et al. “Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles”. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. 2010, pp. 21–26.

[221] Sujatha Das Gollapalli and Cornelia Caragea. “Extracting keyphrases from research papers using citation networks”. In: *Twenty-eighth AAAI conference on artificial intelligence*. 2014.

[222] Thuy Dung Nguyen and Min-Yen Kan. “Keyphrase extraction in scientific publications”. In: *International conference on Asian digital libraries*. Springer. 2007, pp. 317–326.

[223] Anette Hulth. “Improved automatic keyword extraction given more linguistic knowledge”. In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. 2003, pp. 216–223.

[224] Ling Chi and Liang Hu. “ISKE: An unsupervised automatic keyphrase extraction approach using the iterated sentences based on graph method”. In: *Knowledge-Based Systems* 223 (2021), p. 107014.

[225] Zhiyuan Liu et al. “Automatic keyphrase extraction via topic decomposition”. In: *Proceedings of the 2010 conference on empirical methods in natural language processing*. 2010, pp. 366–376.

[226] Florian Boudin. “Unsupervised keyphrase extraction with multipartite graphs”. In: *arXiv preprint arXiv:1803.08721* (2018).

[227] Ricardo Campos et al. “Yake! collection-independent automatic keyword extractor”. In: *European Conference on Information Retrieval*. Springer. 2018, pp. 806–810.

[228] Anna Giabelli et al. “Embeddings evaluation using a novel measure of semantic similarity”. In: *Cognitive Computation* 14.2 (2022), pp. 749–763.

[229] Samuele Colombo et al. “JobSet: Synthetic Job Advertisements Dataset for Labour Market Intelligence”. In: *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*. 2025, pp. 928–935.

[230] CEDEFOP. *Real-time Labour Market information on skill requirements: feasibility study and working prototype". https://goo.gl/qNjmrn*. 2014.

[231] Sergey I Nikolenko. *Synthetic data for deep learning*. Vol. 174. Springer, 2021.

[232] AI@Meta. “Llama 3 Model Card”. In: (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[233] Shitao Xiao et al. “C-Pack: Packaged Resources To Advance General Chinese Embedding”. In: *arXiv preprint arXiv:2309.07597* (2023).

[234] European Organization For Nuclear Research and OpenAIRE. *Zenodo*. en. 2013. DOI: 10.25495/7GXK-RD71. URL: https://www.zenodo.org/.

[235] Niklas Muennighoff et al. “MTEB: Massive Text Embedding Benchmark”. In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 2023, pp. 2014–2037.

[236] Zehan Li et al. “Towards general text embeddings with multi-stage contrastive learning”. In: *arXiv preprint arXiv:2308.03281* (2023).

[237] Sean Lee et al. *Open Source Strikes Bread - New Fluffy Embeddings Model*. 2024. URL: https://www.mixedbread.ai/blog/mxbai-embed-large-v1.

[238] Xianming Li and Jing Li. "AnglE-optimized Text Embeddings". In: *arXiv preprint arXiv:2309.12871* (2023).

[239] Mike Zhang et al. "Skill extraction from job postings using weak supervision". In: *arXiv preprint arXiv:2209.08071* (2022).

[240] Lena Bjerkander and Alexander Glas. "Talking in a language that everyone can understand? Clarity of speeches by the ECB Executive Board". In: *Journal of International Money and Finance* 149 (2024), p. 103200. ISSN: 0261-5606. DOI: https://doi.org/10.1016/j.jimonfin.2024.103200.

[241] Stephen Hansen, Michael McMahon, and Andrea Prat. "Transparency and Deliberation Within the FOMC: A Computational Linguistics Approach*". In: *The Quarterly Journal of Economics* 133.2 (2017), pp. 801–870. DOI: 10.1093/qje/qjx045.

[242] Lorenzo Malandri et al. "Meet: A method for embeddings evaluation for taxonomic data". In: *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2020, pp. 31–38.

[243] Mario Mezzanzanica and Fabio Mercorio. "Big Data Enables Labor Market Intelligence". In: *Encyclopedia of Big Data Technologies*. Ed. by Sherif Sakr and Albert Y. Zomaya. Springer, 2019. DOI: 10.1007/978-3-319-63962-8_276-1. URL: https://doi.org/10.1007/978-3-319-63962-8%5C_276-1.

[244] Nandan Thakur et al. "Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks". In: *arXiv e-prints* (2020), arXiv–2010.

[245] Liang Wang et al. "Improving Text Embeddings with Large Language Models". In: *arXiv preprint arXiv:2401.00368* (2023).

[246] Matt Goldman and David M. Kaplan. "Comparing distributions by multiple testing across quantiles or CDF values". In: *Journal of Econometrics* 206.1 (2018), pp. 143–166. ISSN: 0304-4076.

[247] Lili Shang et al. "A Lexicon Enhanced Collaborative Network for targeted financial sentiment analysis". In: *Information Processing & Management* 60.2 (2023), p. 103187. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2022.103187.

[248] Edward J Hu et al. "Lora: Low-rank adaptation of large language models." In: *ICLR* 1.2 (2022), p. 3.

[249] George Pu et al. *Empirical Analysis of the Strengths and Weaknesses of PEFT Techniques for LLMs*. 2023. arXiv: 2304.14999 [cs.CL]. URL: https://arxiv.org/abs/2304.14999.

[250] Roman Macháček et al. *The Impact of Fine-tuning Large Language Models on Automated Program Repair*. 2025. arXiv: 2507.19909 [cs.SE]. URL: https://arxiv.org/abs/2507.19909.

[251] Scott Barnett et al. *Fine-Tuning or Fine-Failing? Debunking Performance Myths in Large Language Models*. 2024. arXiv: 2406.11201 [cs.CL]. URL: https://arxiv.org/abs/2406.11201.

[252] André Betzer and Jan Philipp Harries. "How online discussion board activity affects stock trading: the case of GameStop". In: *Financial markets and portfolio management* 36.4 (2022), pp. 443–472.

[253] Matthew Gentzkow, Bryan Kelly, and Matt Taddy. "Text as data". In: *Journal of Economic Literature* 57.3 (2019), pp. 535–574.

[254] Tim Loughran and Bill McDonald. "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks". In: *The Journal of finance* 66.1 (2011), pp. 35–65.

[255] Karmvir Singh Phogat et al. "Zero-shot question answering over financial documents using large language models". In: *arXiv preprint arXiv:2311.14722* (2023).

[256] Yinheng Li et al. "Large language models in finance: A survey". In: *Proceedings of the fourth ACM international conference on AI in finance*. 2023, pp. 374–382.

[257] Adam Tauman Kalai et al. "Why Language Models Hallucinate". In: *arXiv preprint arXiv:2509.04664* (2025).

[258] Yazhou Zhang et al. "Sarcasmbench: Towards evaluating large language models on sarcasm understanding". In: *IEEE Transactions on Affective Computing* (2025).

[259] Qing Li et al. "The effect of news and public mood on stock movements". In: *Information Sciences* 278 (2014), pp. 826–840.

[260] Xin Zhang et al. "mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval". In: *arXiv preprint arXiv:2407.19669* (2024).

[261] Sebastian Ruder et al. "Transfer learning in natural language processing". In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*. 2019, pp. 15–18.

[262] Dirk Groeneveld et al. "OLMo: Accelerating the Science of Language Models". In: *Preprint* (2024).

[263] Dahyun Kim et al. *SOLAR 10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling*. 2023. arXiv: 2312.15166 [cs.CL].

[264] Lewis Tunstall et al. *Zephyr: Direct Distillation of LM Alignment*. 2023. arXiv: 2310.16944 [cs.LG].

[265] Neng Wang, Hongyang Yang, and Christina Dan Wang. "Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets". In: *arXiv preprint arXiv:2310.04793* (2023).

[266] Chi Sun et al. "How to fine-tune bert for text classification?" In: *China national conference on Chinese computational linguistics*. Springer. 2019, pp. 194–206.

[267] Jeremy Howard and Sebastian Ruder. "Universal language model fine-tuning for text classification". In: *arXiv preprint arXiv:1801.06146* (2018).

[268] Suchin Gururangan et al. "Don't stop pretraining: Adapt language models to domains and tasks". In: *arXiv preprint arXiv:2004.10964* (2020).

[269] Quinn McNemar. "Note on the sampling error of the difference between correlated proportions or percentages". In: *Psychometrika* 12.2 (1947), pp. 153–157.

[270] Maksym Fedorchuk and Bart Lamiroy. "Binary classifier evaluation without ground truth". In: *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*. IEEE. 2017, pp. 1–6.

[271] Titus J Brinker et al. "Deep neural networks are superior to dermatologists in melanoma image classification". In: *European Journal of Cancer* 119 (2019), pp. 11–17.

[272] Deepshi Garg and Prakash Tiwari. "Impact of social media sentiments in stock market predictions: A bibliometric analysis". In: *Business Information Review* 38.4 (2021), pp. 170–182.

[273] Dennis Huynh et al. "Stock price prediction leveraging reddit: The role of trust filter and sliding window". In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE. 2021, pp. 1054–1060.

[274] Michele Costola, Matteo Iacopini, and Carlo RMA Santagiustina. "On the "mementum" of meme stocks". In: *Economics Letters* 207 (2021), p. 110021.

[275] Andreas M Kaplan and Michael Haenlein. "Users of the world, unite! The challenges and opportunities of Social Media". In: *Business horizons* 53.1 (2010), pp. 59–68.

[276] Farzana Parveen Tajudeen, Noor Ismawati Jaafar, and Sulaiman Ainin. "Understanding the impact of social media usage among organizations". In: *Information & management* 55.3 (2018), pp. 308–321.

[277] Christian Esposito, Vincenzo Moscato, and Giancarlo Sperlí. "Trustworthiness Assessment of Users in Social Reviewing Systems". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.1 (2022), pp. 151–165. DOI: 10.1109/TSMC.2020.3049082.

[278] Christian Esposito, Vincenzo Moscato, and Giancarlo Sperlì. "Detecting malicious reviews and users affecting social reviewing systems: A survey". In: *Computers & Security* 133 (2023), p. 103407. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2023.103407.

[279] Sancheng Peng et al. "Influence analysis in social networks: A survey". In: *Journal of Network and Computer Applications* 106 (2018), pp. 17–32.

[280] Sigit Priyanta, I Nyoman Prayana Trisna, and Nyoman Prayana. "Social network analysis of twitter to identify issuer of topic using pagerank". In: *Int. J. Adv. Comput. Sci. Appl* 10.1 (2019), pp. 107–111.

[281] Armielle Noulapeu Ngaffo, Walid El Ayeb, and Zied Choukair. "Mining user opinion influences on twitter social network: find that friend who leads your opinion using Bayesian method and a new emotional PageRank algorithm". In: *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE. 2019, pp. 680–685.

[282] Huan Zhao et al. "Ranking users in social networks with motif-based pagerank". In: *IEEE Transactions on Knowledge and Data Engineering* 33.5 (2019), pp. 2179–2192.

[283] Jenny L Davis and Timothy Graham. "Emotional consequences and attention rewards: The social effects of ratings on Reddit". In: *Information, Communication & Society* 24.5 (2021), pp. 649–666.

[284] Daniel Loureiro et al. "TimeLMs: Diachronic language models from Twitter". In: *arXiv preprint arXiv:2202.03829* (2022).

[285] Jose Camacho-Collados et al. "TweetNLP: Cutting-edge natural language processing for social media". In: *arXiv preprint arXiv:2206.14774* (2022).

[286] Maurice G Kendall. "A new measure of rank correlation". In: *Biometrika* 30.1-2 (1938), pp. 81–93.

[287] Ke Liang et al. "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (2024), pp. 9456–9478.

[288] Wuzhida Bao et al. "Data-driven stock forecasting models based on neural networks: A review". In: *Information Fusion* 113 (2025), p. 102616.

[289] Soheila Mehrmolaei and Mohammad Saniee Abadeh. "A decade systematic review of fusion techniques in financial market prediction". In: *Computer Science Review* 58 (2025), p. 100813.

[290] Dawei Cheng et al. "Financial time series forecasting with multi-modality graph neural network". In: *Pattern Recognition* 121 (2022), p. 108218.

[291] Linyi Yang et al. "Html: Hierarchical transformer-based multi-task learning for volatility prediction". In: *Proceedings of The Web Conference 2020*. 2020, pp. 441–451.

[292] Jiahao Qin. "Msmf: Multi-scale multi-modal fusion for enhanced stock market prediction". In: *arXiv preprint arXiv:2409.07855* (2024).

[293] Saeede Anbaee Farimani, Majid Vafaei Jahan, and Amin Milani Fard. "An Adaptive Multimodal Learning Model for Financial Market Price Prediction". In: *IEEE Access* (2024).

[294] Jinsong Liu. "Multimodal Data-Driven Factor Models for Stock Market Forecasting". In: *Journal of Computer Technology and Software* 4.2 (2025).

[295] Pinyu Chen, Zois Boukouvalas, and Roberto Corizzo. "A deep fusion model for stock market prediction with news headlines and time series data". In: *Neural Computing and Applications* 36.34 (2024), pp. 21229–21271.

[296] Kun Ouyang et al. "Modal-adaptive knowledge-enhanced graph-based financial prediction from monetary policy conference calls with LLM". In: *arXiv preprint arXiv:2403.16055* (2024).

[297] Ramit Sawhney et al. "FAST: Financial news and tweet based time aware network for stock trading". In: *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*. 2021, pp. 2164–2175.

[298] Hao Qian et al. "Mdgnn: Multi-relational dynamic graph neural network for comprehensive and dynamic stock investment prediction". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 13. 2024, pp. 14642–14650.

[299] Yu Zhao et al. "Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks". In: *IEEE transactions on knowledge and data engineering* 35.8 (2022), pp. 8559–8571.

[300] Tim Bollerslev. "Generalized autoregressive conditional heteroskedasticity". In: *Journal of econometrics* 31.3 (1986), pp. 307–327.

[301] Ibomoiye Domor Mienye, Theo G Swart, and George Obaido. "Recurrent neural networks: A comprehensive review of architectures, variants, and applications". In: *Information* 15.9 (2024), p. 517.

[302] Jongseon Kim et al. "A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges". In: *Artificial Intelligence Review* 58.7 (2025), pp. 1–95.

[303] Ana Lazcano, Pedro Javier Herrera, and Manuel Monge. "A combined model based on recurrent neural networks and graph convolutional networks for financial time series forecasting". In: *Mathematics* 11.1 (2023), p. 224.

[304] Junran Wu et al. "Price graphs: Utilizing the structural information of financial time series for stock prediction". In: *Information Sciences* 588 (2022), pp. 405–424.

[305] Lucas Lacasa et al. "From time series to complex networks: The visibility graph". In: *Proceedings of the National Academy of Sciences* 105.13 (2008), pp. 4972–4975.

[306] Dawei Cheng et al. "Knowledge graph-based event embedding framework for financial quantitative investments". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020, pp. 2221–2230.

[307] Manzhu Yu et al. "Spatiotemporal event detection: A review". In: *International Journal of Digital Earth* 13.12 (2020), pp. 1339–1365.

[308] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[309] Thomas Fischer and Christopher Krauss. "Deep learning with long short-term memory networks for financial market predictions". In: *European journal of operational research* 270.2 (2018), pp. 654–669.

[310] Wei Bao, Jun Yue, and Yulei Rao. "A deep learning framework for financial time series using stacked autoencoders and long-short term memory". In: *PloS one* 12.7 (2017), e0180944.

[311] Yong Yu et al. "A review of recurrent neural networks: LSTM cells and network architectures". In: *Neural computation* 31.7 (2019), pp. 1235–1270.

[312] Akhil Sethia and Purva Raut. "Application of LSTM, GRU and ICA for stock price prediction". In: *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 2*. Springer, 2018, pp. 479–487.

[313] Pushpendu Ghosh, Ariel Neufeld, and Jajati Keshari Sahoo. "Forecasting directional movements of stock prices for intraday trading using LSTM and random forests". In: *Finance Research Letters* 46 (2022), p. 102280.

[314] Jian Cao, Zhi Li, and Jian Li. "Financial time series forecasting model based on CEEMDAN and LSTM". In: *Physica A: Statistical mechanics and its applications* 519 (2019), pp. 127–139.

[315] Yao Qin et al. "A dual-stage attention-based recurrent neural network for time series prediction". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2017, pp. 2627–2633.

[316] Hossein Abbasimehr and Reza Paki. "Improving time series forecasting using LSTM and attention models". In: *Journal of Ambient Intelligence and Humanized Computing* 13.1 (2022), pp. 673–691.

[317] Franco Scarselli et al. "The graph neural network model". In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

[318] Yi-Ling Hsu, Yu-Che Tsai, and Cheng-Te Li. "Fingat: Financial graph attention networks for recommending top-*k* k profitable stocks". In: *IEEE transactions on knowledge and data engineering* 35.1 (2021), pp. 469–481.

[319] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[320] Yunhua Pei, Jin Zheng, and John Cartlidge. "Dynamic Graph Representation with Contrastive Learning for Financial Market Prediction: Integrating Temporal Evolution and Static Relations". In: *arXiv preprint arXiv:2412.04034* (2024).

[321] Jianliang Gao et al. "Graph-based stock recommendation by time-aware relational attention network". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16.1 (2021), pp. 1–21.

[322] Nancy Xu, Chrysoula Kosma, and Michalis Vazirgiannis. "TimeGNN: temporal dynamic graph learning for time series forecasting". In: *International Conference on Complex Networks and Their Applications*. Springer. 2023, pp. 87–99.

[323] Yuhan Wang. "Stock Prediction with Improved Feedforward Neural Networks and Multimodal Fusion". In: *Journal of Computer Technology and Software* 4.1 (2025).

[324] Ruirui Liu et al. "Multimodal multiscale dynamic graph convolution networks for stock price prediction". In: *Pattern Recognition* 149 (2024), p. 110211.

[325] Qing Li et al. "A multimodal event-driven LSTM model for stock prediction using online news". In: *IEEE Transactions on Knowledge and Data Engineering* 33.10 (2020), pp. 3323–3337.

[326] Ziniu Hu et al. "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction". In: *Proceedings of the eleventh ACM international conference on web search and data mining*. 2018, pp. 261–269.

[327] Gary Ang and Ee-Peng Lim. "Guided attention multimodal multitask financial forecasting with inter-company relationships and global and local news". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 6313–6326.

[328] Rui Cheng and Qing Li. "Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 1. 2021, pp. 55–62.

[329] Shuqi Li et al. "PEN: prediction-explanation network to forecast stock price movement with better explainability". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 4. 2023, pp. 5187–5194.

[330] Pulikandala Nithish Kumar, Nneka Umeorah, and Alex Alochukwu. "Dynamic graph neural networks for enhanced volatility prediction in financial markets". In: *arXiv preprint arXiv:2410.16858* (2024).

[331] Ramit Sawhney et al. "Deep attentive learning for stock movement prediction from social media text and company correlations". In: *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*. 2020, pp. 8415–8426.

[332] Wentao Xu et al. "Rest: Relational event-driven stock trend forecasting". In: *Proceedings of the web conference 2021*. 2021, pp. 1–10.

[333] Shibo Feng et al. "Relation-aware dynamic attributed graph attention network for stocks recommendation". In: *Pattern Recognition* 121 (2022), p. 108119.

[334] Liang Zhao et al. "Long-term, short-term and sudden event: trading volume movement prediction with graph-based multi-view modeling". In: *arXiv preprint arXiv:2108.11318* (2021).

[335] Wei Li et al. "Modeling the stock relation with graph network for overnight stock movement prediction". In: *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 2021, pp. 4541–4547.

[336] Xiurui Hou et al. "St-trader: A spatial-temporal deep neural network for modeling stock market movement". In: *IEEE/CAA Journal of Automatica Sinica* 8.5 (2021), pp. 1015–1024.

[337] Oren Etzioni et al. "Open information extraction from the web". In: *Communications of the ACM* 51.12 (2008), pp. 68–74.

[338] Hiroaki Sakoe. "Dynamic-programming approach to continuous speech recognition". In: *1971 Proc. the International Congress of Acoustics, Budapest*. 1971.

[339] Dawei Zhou et al. "Domain adaptive multi-modality neural attention network for financial forecasting". In: *Proceedings of The Web Conference 2020*. 2020, pp. 2230–2240.

[340] Yong Shi et al. "Integrated gcn-lstm stock prices movement prediction based on knowledge-incorporated graphs construction". In: *International Journal of Machine Learning and Cybernetics* 15.1 (2024), pp. 161–176.

[341] Sheng Xiang et al. "Temporal and heterogeneous graph neural network for financial time series prediction". In: *Proceedings of the 31st ACM international conference on information & knowledge management*. 2022, pp. 3584–3593.

[342] Ramit Sawhney et al. "Multimodal multi-task financial risk forecasting". In: *Proceedings of the 28th ACM international conference on multimedia*. 2020, pp. 456–465.

[343] Hung-Yang Li, Vincent S Tseng, and S Yu Philip. "Enhancing stock trend prediction models by mining relational graphs of stock prices". In: *2020 International Conference on Pervasive Artificial Intelligence (ICPAI)*. IEEE. 2020, pp. 110–117.

[344] Jiexia Ye et al. “Multi-graph convolutional network for relationship-driven stock movement prediction”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 6702–6709.

[345] Sang Il Lee and Seong Joon Yoo. “Multimodal deep learning for finance: integrating and forecasting international stock markets”. In: *The Journal of Supercomputing* 76 (2020), pp. 8294–8312.

[346] Shibal Ibrahim, Max Tell, and Rahul Mazumder. “Dyn-GWN: Time-Series Forecasting using Time-varying Graphs with Applications to Finance and Traffic Prediction”. In: *Proceedings of the Fourth ACM International Conference on AI in Finance*. 2023, pp. 167–175.

[347] Harald Cramér. *Mathematical methods of statistics*. Vol. 9. Princeton university press, 1999.