Saint Petersburg State University

Tomin Denis Valerievich

Bachelor Diploma Thesis

Understanding the Aspect Structure of Financial Publications Using Deep Neural Networks

Level of education: Bachelor's degree
Direction 01.03.02 "Applied Mathematics and Informatics"
Basic educational program CB.5005.2015 «Management»
Graduated School of Management

Supervisor:
Professor, Research Center for Market Efficiency and Applied Finance,
Dr. Darko Vuković

Peer reviewer:
Senior Lecturer, Department of Finance and Accounting,
Vitaly Leonidovich Okulov

Saint Petersburg
2025

Contents

Introduction

In recent years, the use of data analytics and artificial intelligence, specifically machine learning (ML) and deep learning (DL), to make investment decisions has become an integral part of many companies' and funds' strategies. However, today's financial markets are characterized by high volatility and high speed of information dissemination, which creates significant challenges for analyzing its impact on stock prices.

Events such as news releases, regulatory changes and analysts' reviews can have both immediate and cumulative effects on market performance. However, traditional approaches to analyzing data often ignore the dynamics of these influences, resulting in poor forecast accuracy and, consequently, ineffective investment strategies.

To compete in a rapidly changing financial environment, companies need to continuously optimize their approaches to data analysis. This requires the development of tools that can not only account for the dynamic nature of information, but also provide forecasts based on an in-depth analysis of events and their cumulative effect. This paper responds to these challenges by proposing a methodology and a technological solution for more accurate stock price forecasting based on deep neural networks, namely large language models (LLM).

Classical ML algorithms have demonstrated their effectiveness in financial forecasting in numerous studies. However, DL and natural language processing (NLP) architectures have fundamentally shifted the paradigm following the emergence of the Transformer architecture in 2017 [Vaswani (et al.), 2017]. Since then, LLMs have gained wide acceptance and proven their applicability across various applied tasks, including asset price forecasting [Halder, 2022; Jiang, Zeng, 2023; J. Kim, H. S. Kim, Choi, 2023].

Contemporary research demonstrates the high efficacy of LLMs in addressing a range of tasks related to asset evaluation and forecasting. Nevertheless, unresolved issues remain regarding the integration of LLMs with classic quantitative models, the scarcity of open-source solutions for the financial domain, and the limitations of current models in processing long textual sequences (see Section 1.3.1). In December 2024, a new state-of-the-art (SoTA) model, ModernBERT, was introduced, capable of processing texts that are 16 times longer than those handled by previous architectures [Devlin (et al.), 2019; Warner (et al.), 2024]. This model extends analytical capabilities from isolated headlines or posts to full news articles, press releases, interview transcripts, and analytical reviews. Nonetheless, processing comprehensive financial reports (e.g., 10-Q, 10-K) remains a challenging task (see Section 2.1.4). Moreover, focusing on full articles helps to mitigate issues related to clickbait and insufficient context.

Pre-trained language models on general text corpora do not always effectively address financial forecasting tasks [Jiang, Zeng, 2023]. This is due to the unique nature of financial information, characterized by specialized terminology and jargon, which complicates the application of general-purpose models. Consequently, there is a need to adapt baseline LLMs for the financial domain.

From a management perspective, when making investment decisions in volatile markets, it is crucial to promptly analyze the cumulative impact of various events (news, regulatory changes, analyses, etc.) on asset dynamics. Experts are unable to process such an immense volume of information within extremely short time frames. Conversely, the absence of a comprehensive analysis tool leads to delayed or inaccurate decisions, reducing the effectiveness of investment strategies and increasing the risk of missing lucrative opportunities.

The development of such a tool is complex and demanding, and it can be conceptually divided into the following stages:

1. Development of an effective architecture for LLMs.
2. Adaptation of the model to the specifics of the financial domain.
3. Fine-tuning of the model to address particular tasks.
4. Integration of the model into a system operating with both quantitative and qualitative data, encompassing training, testing, and deployment processes.

Since the basic architecture of ModernBERT has already been established, the present study focuses on its domain adaptation. Among the available adaptation methods — fine-tuning, complete retraining, and domain-adaptive pretraining — the latter is emphasized in this work (see Section 1.3.2), as it minimizes time, computational, and financial costs.

Within the scope of this research, hierarchical aspect-based analysis of financial publications (see Section 3.2) is considered an effective task for financial forecasting. The object of the study is investment strategies based on the use of artificial intelligence, while the subject is the integration of language models into asset price forecasting processes. The goal of this work is to develop a practice-oriented toolkit for dynamic multimodal forecasting using aspect-based sentiment analysis. It is important to emphasize that a key requirement for the proposed solution is its interpretability, in contrast to other approaches that employ deep neural networks as black-box models.

Throughout the study, the following key tasks were undertaken:

- Selection and analysis of SoTA architectures and models (see Section 1.3.1).
- Investigation of cutting-edge techniques and algorithms to enhance the performance of deep neural networks (see Section 1.3.2).
- Evaluation of various metrics, datasets, and benchmarks to assess the efficiency of the final solution (see Section 1.4).
- Determination of the technology stack for developing the solution (see Section 1.5).

In total, more than 6 GB of exclusively financial texts were collected for the domain adaptation of ModernBERT (see Section 2.2.2). Following comprehensive analysis and preprocessing of the data (see Sections 2.2.3 and 2.2.4, respectively), the domain adaptation of the model was performed (see Section 2.3) and inference was conducted on the task of hierarchical clustering of texts (see Section 2.4). The final outcome includes a comparison of key benchmarks between the original and the domain-adapted models (see Section 3.1), analysis and interpretation of the results obtained from the hierarchical clustering of financial publications (see Section 3.2), as well as the development of a multimodal architecture for dynamic asset forecasting based on aspect-based sentiment analysis (see Section 3.3) and a mathematical framework for semantic deduplication of texts (see Section 3.4).

All the results of this study, including data collection code, training code, analyses, and results are available in the official repository of the project[1]. The domain-adapted model[2] and the collected corpus[3] are also publicly available.

[1]URL: `https://github.com/denisalpino/FinABYSS`
[2]URL: None
[3]URL: `https://huggingface.co/datasets/denisalpino/YahooFinanceNewsRaw`

CHAPTER 1. THEORETICAL OVERVIEW

1.1 Artificial Intelligance in Finance

1.1.1 Price Prediction

There exists a variety of approaches demonstrating the effectiveness of asset price prediction using deep neural networks. Recurrent neural networks (RNN) are undeniably the most prevalent for this task, while convolutional neural networks (CNN) are often employed as auxiliary components.

The long short-term memory (LSTM) [Hochreiter, Schmidhuber, 1997] architecture is the most representative member of the RNN family and is frequently applied to price forecasting. Its adaptations — such as bidirectional LSTM (Bi-LSTM) and hybrid LSTM+CNN models — also exhibit high performance. Furthermore, with the advent of the Transformer [Vaswani (et al.), 2017] architecture, research has increasingly focused on adapting it to the characteristics of time series [Wen (et al.), 2022].

Among recent experiments, the following are particularly noteworthy:

- A repository demonstrating the potential of the Transformer architecture for Bitcoin price forecasting[4];
- A study comparing the performance of Bi-LSTM, hybrid Bi-LSTM+CNN, and Transformer models in predicting IBM stock prices[5];
- An LSTM-based trading bot designed to capture short-term profits during sideways price movements of the CGEN asset, which achieved a 4% daily return on deposit in backtesting[6].

Nevertheless, individual models — whether LSTM-based or decision tree–based — have limited adaptability to changing market regimes and struggle to adjust to their dynamics [Vuković (et al.), 2024]. Although this limitation is evident, it is often underestimated in the literature. According to the Efficient Market Theory (EMT) [Fama, 1970], asset prices incorporate all available market information, which casts doubt on the feasibility of precise forecasting using only historical quantitative indicators (OHLC data, trading volume, and classical technical indicators).

Moreover, most models fail to account for nuanced factors such as limit order book dynamics, interactions with other trading algorithms, and qualitative off-exchange information. Recent studies indicate that integrating analysis of the off-exchange information environment into predictive models significantly enhances forecast accuracy, as will be detailed in the subsequent sections.

[4]URL: `https://github.com/baruch1192/-Bitcoin-Price-Prediction-Using-Transformers`
[5]URL: `https://github.com/JanSchm/CapMarket`
[6]URL: `https://github.com/roeeben/Stock-Price-Prediction-With-a-Bot`

1.1.2 Sentiment Analysis

As noted in the Introduction, the emergence of the Transformer architecture has enabled deep learning models to achieve significant progress in natural language understanding (NLU). This capability is of particular value in finance, where traditional quantitative data alone prove insufficient for precise forecasting.

Prior to the widespread adoption of modern language models for sentiment analysis (SA), various approaches were employed — LSTM–based networks, ULMFiT, autoregressive architectures, and others [Hochreiter, Schmidhuber, 1997; Howard, Ruder, 2018]. For example, one study compared an autoregressive model without sentiment features to an otherwise identical model augmented with sentiment inputs; it demonstrated that in 77.8% of cases the sentiment–aware version outperformed its purely quantitative counterpart [Vanstone, Gepp, Harris, 2019].

Modern pre-trained Transformer models, commonly referred to as large language models (LLMs), open new horizons for financial forecasting. The most intuitive application is the SA of news. This process requires the collection of a substantial corpus of texts, which are then annotated with one of three labels: -1 (negative), 0 (neutral), or 1 (positive) [Pathak (et al.), 2020].

Manual annotation typically involves domain experts capable of assessing a text's impact on financial markets. To ensure high data quality, the cross-consensus method [Bogdan, Biklen, 1997] is often applied: multiple experts independently label the same texts, and their labels are reconciled. This methodology underpinned the creation of the widely used FinancialPhraseBank dataset for fine-tuning models to financial sentiment analysis (FSA) tasks [Malo (et al.), 2014].

Algorithmic labeling methods based on asset price dynamics are less common, owing to their subjectivity and instability. In particular, it is difficult to define a reliable pric–change threshold for sentiment classification and to guarantee that observed gains are not driven by extraneous factors.

Once annotation is complete — a process consuming the majority of resources — pre-trained language models are fine-tuned by adjusting the weights of the final neural layers.

The SA taxonomy comprises three levels [Pathak (et al.), 2020]:

- **Document level.** Sentiment is evaluated across an entire document (e.g., a news article or report). This level assumes opinions pertain to a single entity, which seldom reflects reality. Furthermore, LLMs are technically constrained by maximum token counts and therefore cannot process multi-page filings (such as 10-K reports) or lengthy news articles in full. Recent advances have improved the handling of standard news articles, yet comprehensive reports remain beyond reach.

- **Sentence level.** Sentiment is assigned to individual sentences or short passages (typically 1–3 sentences), still under the assumption of a single subject. Most SA research operates at this level, supported by abundant datasets and well within the token processing capacity of modern LLMs. Common sources include news headlines and social-media posts on

platforms such as X (formerly Twitter) and Reddit.

- **Aspect level (ABSA).** This level captures sentiment toward specific aspects of an entity. ABSA is the most advanced SA tier and is treated as a distinct field encompassing four subtasks: aspect extraction, aspect-polarity assignment, aspect-category detection, and category-polarity assignment. A specialized variant, targeted ABSA, permits multiple entities but restricts sentiment to one per entity. Further discussion of ABSA appears in Section 1.1.3.

Most SA methodologies require extensive text annotation, which represents a major drawback due to high resource demands and inherent limitations:

- No universal, precise definition of "sentiment" fits every application, forcing bespoke definitions for each task — a challenge given the diversity of textual patterns and contexts.
- Without significant investments in quality control (e.g., cross-consensus [Bogdan, Biklen, 1997]), it is impossible to guarantee annotation reliability and objectivity, since human factors heavily influence label consistency.

As mentioned, SA's formulation may vary across domains. FSA is distinguished by its reliance not only on textual data but also on the abundant quantitative information present in financial publications [Du (et al.), 2024].

Nevertheless, FSA encounters failures in several common scenarios, including: irrealis moods (conditional, subjunctive or imperative moods), rhetorical devices (negative assertions, personification, sarcasm), dependent opinions, unspecified aspects, out-of-vocabulary terms (jargon, microtext, named entities), external references (allusions to knowledge not encoded in the model) [Xing (et al.), 2020].

1.1.3 Aspect Analysis

«To be done later»

1.2 Machine Learning Algorithms

1.2.1 Dimensionality Reduction

Principal Component Analysis (PCA) is a linear dimensionality reduction method based on the decomposition of the covariance matrix of the original data. Given a zero trait mean, it finds the covariance eigenvectors (components) responsible for the maximum variance: $\Sigma = X^T X$, $\Sigma w_i = \lambda_i w_i$, the projection of the data $Y = XW_{d'}$ onto the first d' eigenvectors.

PCA aims to preserve as much variance of the original data as possible and, as a consequence, preserves well the "global" cluster structure [Amid, Warmuth, 2019]. In practice, PCA

is often used as an intermediate step: for example, when dealing with high-dimensional text embeddings (several hundred features, e.g., embeddings of dimensionality 768) before non-linear dimensionality reduction methods. The use of PCA can significantly reduce the dimensionality (down to tens to hundreds of components) and speed up subsequent computations[H. Huang (et al.), 2022].

However, PCA is a linear method, and it does not capture the more complex nonlinear dependencies inherent in language embeddings. As a consequence, subtle local word-document relationships may be lost, although the overall structure (the "global landscape" of the data) is most often preserved.

t-distribution Stochastic Meighbor Embedding (t-SNE) is a nonlinear stochastic method focused on preserving local data structure. In high-dimensional space, it computes conditional probabilities $p_{j|i} \propto \exp(-|x_i - x_j|^2/2\sigma_i^2)$, reflecting the proximity of neighbors; it then symmetrizes them: $p_{ij} = (p_{j|i} + p_{i|j})/2n$. A similar measure (Student's distribution with 1 degree of freedom) is given in the low-dimensional mapping. The algorithm optimizes the placement of Y points by minimizing the Kullback-Leibler divergence $C = \sum_{i,j} p_{ij} \ln \frac{p_{ij}}{q_{ij}}$. As a result, points close to each other in the original space will retain a local cluster structure in the low-dimensional space.

The 'perplexity' hyperparameter determines the number of effective neighbors. t-SNE shows an impressive visualization of local clusters, but poorly reproduces the global distance between clusters. It is computationally expensive for large samples and is usually used only for the final transition to 2D space (or 3D), not for intermediate dimensionality reduction. For textual embeddings, t-SNE is often applied after preprocessing (e.g., PCA dimensionality reduction) because it scales poorly directly to several hundred features.

Uniform Manifold Approximation and Projection (UMAP) is a method of nonlinear dimensionality reduction based on the manifold assumption. UMAP theoretically relies on Riemannian geometry and the theory of "fuzzy simplicial sets". The algorithm constructs a graph of k-nearest neighbors in the original space, then each pair of points is assigned a "membership" in a fuzzy set by a formula of the form:

$$\mu_{ij} = \exp\big(-\frac{\max(0, d(x_i, x_j) - \rho_i)}{\sigma_i}\big), \tag{1}$$

where ρ_i takes into account the density of neighbors. These local symplectic sets are then combined and symmetrized, and a weighted data graph is obtained. Next, a similar "fuzzy" graph is constructed in low-dimensional space and the location of points is optimized by minimizing the cross entropy energy between the two graphs.

UMAP retains the local structure of the data while aiming to distribute points uniformly on the manifold; unlike t-SNE, it can better retain some global features (due to the cross-entropy used). The main hyperparameters of UMAP are the number of neighbors 'n_neighbors' (specifies the scale of locality) and 'min_dist' (minimum distance of points in the mapping).

UMAP demonstrates high speed and scalability (the method can run on an arbitrary number of output measurements at once). In an experiment, UMAP was shown to give a visualization quality comparable or better than t-SNE with a significantly shorter runtime [McInnes, Healy, Melville, 2018]. UMAP's advantages also include the preservation of a larger-scale data structure and the ability to "downscale" the dimensionality down to multidimensional vectors (not only 2D).

Nevertheless, UMAP may incorrectly represent highly sparse clusters by "equalizing" dense and sparse regions (the algorithm actually seeks a uniform distribution of data on the assumed manifold). In addition, UMAP results are sensitive to the choice of hyperparameters and the degree of noise sampling (the algorithm uses approximate neighbor search and negative sample sampling) [H. Huang (et al.), 2022]. However, in most textual data clustering applications, UMAP has proven to be a robust and efficient tool.

There are several Python implementations of this method. The classical implementation, 'umap-learn' (CPU), is widely used [McInnes, Healy, Saul, (et al.), 2018], and there is a GPU implementation ('cuML') to speed up the learning process [Raschka, Patterson, Nolet, 2020]. cuML's GPU implementation of UMAP can give speedups of up to 10-100× compared to the CPU version on large amounts of data. However, in early versions of cuML, approximations were introduced for speed purposes, sometimes resulting in a small difference in display quality compared to the original. Also the GPU implementation of UMAP may reduce the local structure preservation relative to the reference version. Thus, 'cuML' UMAP is suitable for very large amounts of data, while the implementation from 'umap-learn' is more versatile and stable, but runs slower.

Triplet Manifold Approximation and Projection (TriMAP) is an embedding learning method with an emphasis on the global [Amid, Warmuth, 2019] data structure. It formulates the problem through triples of (i, j, k) points: point i must be closer to j than to k in a low-dimensional representation. The selection of such triplets is based on nearest and farthest neighbors in the original space; each triplet is given a weight reflecting the relative closeness of the pairs in the original space. Optimization is performed over a large sample of informative triplets using gradient descent.

TriMAP preserves global structure much better than t-SNE and often better than UMAP. TriMAP also scales well and exhibits low runtime for large and high-dimensional samples [Ibid.]. In terms of text embeddings, TriMAP can provide a more readable picture of document cluster locations on 2D (although local community details may be smoothed out).

Pairwise Controlled Manifold Approximation and Projection (PaCMAP) is a newer method specifically designed to balance local and global structure [Y. Wang (et al.), 2021]. Like TriMAP, it uses samples of pairs of points of different types: "close" pairs (neighbors), "middle" pairs (between clusters), and "far" pairs. For each pair type, the corresponding weights and attraction/repulsion forces are specified. As a result, the optimized loss function tends to simultaneously compress locally close points and push distant ones apart, preserving the global shape of the distribution.

PaCMAP is robust to hyperparameter selection and dimensionality reduction in preprocessing, and preserves both local and global structure well. The disadvantages of PaCMAP are its relative novelty and the need to fit fractions of different types of pairs.

Systematic comparisons indicate a characteristic separation in the properties of these algorithms. Thus, PCA, TriMAP, and PaCMAP preserve global distances (large-scale cluster structure) well, while t-SNE and UMAP are better at capturing local details [H. Huang (et al.), 2022]. PCA is traditionally used for preprocessing: reducing the dimensionality to tens of components speeds up further analysis and makes it more stable. However, it is noticeable that full PCA preprocessing can distort the original distances, so the results of the final embedding (e.g., t-SNE/UMAP visualization) often depend on the number of PCA components. In experimental method evaluations, PaCMAP and TriMAP showed the best agreement of global distances, while UMAP and t-SNE were on average inferior in this task. Conversely, t-SNE and UMAP performed best in the classification task on vector features (testing local consistency).

In general, the choice of dimensionality reduction method for high-dimensional embeddings of documents depends on the task: for subsequent clustering and thematic one often uses PCA or UMAP (for more "stable" representation of clusters), and for final 2D visualization and detailed analysis of local clusters — t-SNE, UMAP or PaCMAP. Careful selection of hyperparameters and possibly a combination of methods (e.g., PCA+UMAP) can achieve a better representation of the textual data structure.

1.2.2 Clustrering

In the considered scheme for thematic analysis of financial news articles, a dimensionality reduction algorithm is applied after obtaining textual embeddings. After this dimensionality reduction, clustering algorithms operate in a less sparse space, which can improve the selection of dense regions and reduce the influence of noise factors.

K-Means is one of the classical partitional clustering algorithms [Jain, 2010]. It seeks to partition the data into K clusters by minimizing the intra-cluster point spread. We denote clusters $C_1, \ldots, C_K$ and cluster centroids μ_k; then the optimized objective function (the "sum of squares of distances" metric from points to the corresponding centroids) is given as

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2, \tag{2}$$

and is minimized when partitioned by the Euclidean metric. Finding the global minimum of this function is an NP-complete problem, so the K-Means method performs a greedy iterative procedure of point reclassification and centroid recalculation (usually with random initialization) that converges to a local minimum. An important feature of K-Means is the requirement to specify the number of clusters K and the initial approximations in advance.

Since the algorithm typically uses a Euclidean metric, it forms mostly spherical clusters.

All objects are automatically assigned to some cluster (rigidly "belonging" each point to one cluster), and the algorithm does not explicitly emphasize outliers or noise. Advantages of the method include ease of implementation, low computational cost, and widespread use. However, K-Means is unstable to outliers, and is poor at distinguishing nested or highly heterogeneous clusters.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering method [Ester (et al.), 1996]. It defines clusters as regions of high-density data separated by low-density regions. The algorithm uses two parameters: the radius ε and the minimum number of points min_{pts}. A point is called the "core" of a cluster if its ε-neighborhood contains at least min_{pts} points. Points reachable in density from the core belong to the same cluster.

DBSCAN automatically separates points that are not in dense regions as noise and does not require the number of clusters to be specified. Due to this, the method finds clusters of arbitrary shape and is well suited for datasets with non-uniformly distributed objects. However, DBSCAN has significant limitations: the choice of a single ε threshold is critical, and when combining clusters of different densities, the algorithm either merges them into one whole or breaks them into too small fragments. In addition, as the dimensionality of the space grows, the data becomes sparse, and it becomes difficult to distinguish a high-density region from a low-density one. As a consequence, DBSCAN in high-dimensional text embeddings often demonstrates reduced efficiency. Also, the complexity of classical DBSCAN in the absence of optimization can reach $O(n^2)$, although practical implementations with indexes usually have significantly lower performance.

Hierarchical DBSCAN (HDBSCAN) is a hierarchical extension of the DBSCAN method [Campello, Moulavi, Sander, 2013]. Unlike DBSCAN, HDBSCAN does not require a fixed ε: instead, it computes the mutual reachability distance between points x and y as

$$d_{\text{mreach}}(x,y) = \max\{d_{\text{core}}(x), d_{\text{core}}(y), d(x,y)\}, \tag{3}$$

where $d_{\text{core}}(x)$ is the distance from a point x to its k-th nearest neighbor (i.e., the minimum ε at which x becomes the "core" of the cluster). We then construct a graph of complete mutual reachability (or directly a minimal island tree at such distances). By removing the edges of this tree in descending order of weight, the algorithm generates a tree-like hierarchy of clusters reflecting the nested structure of the data at different density thresholds. From this tree, the final clusters are selected based on the stability criterion. Such a procedure is equivalent to performing multiple runs of DBSCAN at all possible ε and selecting the most significant "stable" clusters. An important feature of HDBSCAN is that it finds the optimal number of clusters by itself, requiring only a minimum cluster size ('min_cluster_size').

Thanks to its hierarchical approach, HDBSCAN is able to detect nested clusters of different densities and adapt more flexibly to the data distribution than DBSCAN and K-Means. The algorithm is robust to density fluctuations: if there are regions of different homogeneity in the data, HDBSCAN will identify large sparse clusters and smaller dense clusters simultaneously. It

automatically flags outliers, similar to DBSCAN, but without rigidly binding to a single threshold. Because of the additional processing (searching for k-nearest neighbors, MST construction, and hierarchy analysis), HDBSCAN is somewhat more computationally complex, but modern implementations with efficient neighborhood structures usually provide comparable or even better performance [McInnes, Healy, Astels, 2017]. In general, HDBSCAN provides a richer description of the data structure at different levels of granularity and more often yields more meaningful clusters in complex multidimensional spaces.

Existing experimental studies show that in the task of topic analysis of long texts, each method has its own pros and cons. K-Means is often used as a basic method due to its simplicity and scalability, but it gives a relatively coarse partitioning of topics because it is limited by the spherical shape of clusters and requires the number of topics to be specified in advance. DBSCAN can detect arbitrarily shaped clusters and separate noise, but its performance is reduced on high-dimensional embeddings of texts due to sparsity and the need to tune the global density threshold.

In many comparative experiments, HDBSCAN is shown to outperform both mentioned methods in terms of the quality of topic clusters: it automatically adapts to the variability of embedding density, detects nested topics of different granularity, and reliably eliminates irrelevant noise [Campello, Moulavi, Sander, 2013; McInnes, Healy, Astels, 2017]. These findings are supported by practical applications (e.g., news article clustering), where HDBSCAN most often yields more interpretable and stable results compared to K-Means or "flat" DBSCAN [Grootendorst, 2022].

1.2.3 Evaluation

Silhouette Index + Stability Index + DBCV

1.3 Deep Neural Networks

1.3.1 Models

LSTM [Hochreiter, Schmidhuber, 1997]

«...»

BERT [Devlin (et al.), 2019]

«...»

FinBERT (2019 — first) [Araci, 2019]

«...»

FinBERT (2020 — good) [A. Huang, H. Wang, Y. Yang, 2023; Y. Yang, UY, A. Huang, 2020]

«...»

FinBERT (2020 — best) [Liu (et al.), 2020]

«...»

ModernBERT [Warner (et al.), 2024]

«...»

1.3.2 Techniques

Domain-Adaptive Pretraining (DAPT). — [Gururangan (et al.), 2020]

Based on the calculations, DAPT provides on average a 4% increase in benchmarks on a relative scale compared to the baseline model, which was not domain-specific. This figure is quite significant, considering that for some specific tasks the gains can be as high as 20% [Ibid.].

Fusion Mechanisms. There are three commonly accepted mechanisms for multimodal data fusion.

The first, Early Fusion, involves feeding all features (stock quotes, technical indicators, and textual embeddings) at once into a single CNN-LSTM model [Dutt, Zare, Gader, 2022; Karpathy (et al.), 2014]. The advantage of the method is the ease of implementation and the ability to immediately learn cross-modal dependencies. However, in practice Early Fusion is prone to "choking" in the noise of one of the modalities and loses flexibility when dynamically estimating the contribution of each modality [Dutt, Zare, Gader, 2022].

The second method, Late Fusion, combines the predictions of individual channels (each modality is processed by its CNN-LSTM branch) only at the final stage [Karpathy (et al.), 2014; Ortega (et al.), 2019]. This approach is characterized by modularity (easy replacement or additional training of a single channel), but it excludes extraction of low-level cross-modal patterns and requires training all branches separately, which entails a multiple increase in computational resources [Joze (et al.), 2020].

The third, compromise mechanism, Slow Fusion, provides staged, "slow" merging of links at different layers of the network [Dutt, Zare, Gader, 2022; Feichtenhofer, Pinz, Zisserman, 2016]. Slow Fusion approaches Early Fusion for early merging, and Late Fusion for late merging. The key advantages of the method are: the balance between the autonomous processing of each modality and the possibility of taking into account their interaction, preserving the "purity" of low-level features and the flexibility of setting the number and depth of integration stages [Karpathy (et al.), 2014]. The main disadvantages are the difficulty of choosing the optimal fusion level and increased computational costs due to parallel branches at early layers, but nevertheless, less than in Late Fusion.

Approaches to representation document embedding. [CLS] token and Mean-pooling

1.3.3 Benchmarks

The General Language Understanding Evaluation (GLUE) benchmark constitutes a standardized framework for assessing the language comprehension capabilities of natural language processing (NLP) models [A. Wang (et al.), 2018]. It comprises 9 tasks encompassing classification, semantic similarity evaluation, and textual entailment recognition. Through the diversity

of these tasks, GLUE facilitates the identification of models' ability to generalize and effectively transfer learned representations across a range of linguistic challenges.

The primary components of GLUE include:

- **A suite of nine tasks**, each derived from pre-existing corpora and targeting distinct aspects of language understanding (e.g., linguistic acceptability, sentiment analysis, paraphrase detection).
- **A diagnostic dataset** for an in-depth evaluation of model performance in capturing various linguistic phenomena.
- **A public leaderboard and dashboard** that enable continuous tracking of benchmark performance and provide visualization of model results on the diagnostic tasks.

Below is a summary table of the key characteristics of the datasets included in the GLUE benchmark:

Table 1: Overview of GLUE Benchmark Datasets

Name	Task	Source	Size	Metric
CoLA	Single-Sentence Classification	[Dudy (et al.), 2018]	∼8 500	Matthews Correlation Coefficient
SST-2	Binary Single-Sentence Sentiment Classification	[Socher (et al.), 2013]	∼67 000	Accuracy
MRPC	Paraphrase Identification	[Dolan, Brockett, 2005]	∼3 700	Accuracy, F1
STS-B	Semantic Textual Similarity (Regression)	[Cer (et al.), 2017]	∼7 000	Pearson/Spearman Correlation
QQP	Duplicate Question Detection (Quora Question Pairs)	[Z. Chen (et al.), 2017]	∼364 000	Accuracy, F1
MNLI	Multi-Genre Natural Language Inference	[Williams, Nangia, Bowman, 2018]	∼393 000	Accuracy
QNLI	Inference Tasks	[Rajpurkar (et al.), 2016]	∼105 000	Accuracy
RTE	Recognizing Textual Entailment	[Bentivogli (et al.), 2009]	∼2,500	Accuracy
WNLI	Inference Tasks	[Levesque, Davis, Morgenstern, 2012]	634	Accuracy

In summary, the GLUE benchmark provides a robust foundation for evaluating both standard and domain-adapted NLP models. Its comprehensive design and the inclusion of diverse linguistic tasks allow for a nuanced analysis of model capabilities. Following this overview, the FLUE benchmark, which is tailored for the evaluation of models in the financial context, will be discussed to further complement the assessment of domain-adaptive pre-training strategies.

The Financial Language Understanding Evaluation (FLUE) benchmark is a domain-specific analog to the GLUE benchmark, tailored specifically for the financial domain [Shah (et al.), 2022]. This benchmark was developed very recently based on 5 diverse datasets. Its creation was driven by the need to evaluate models capable of effectively processing financial texts, as standard general-purpose datasets often fail to capture the unique characteristics of financial lexicon and the specific tasks inherent to this domain.

FLUE covers 5 distinct financial tasks, which allow for a comprehensive evaluation of model performance across various aspects of financial language. The statistics presented in Table 2 demonstrate the scale and diversity of the included datasets. Moreover, all datasets that comprise FLUE are characterized by low ethical risks and do not contain confidential information regarding any organization or individual. In addition, explicit consent was obtained from the authors of each dataset prior to their inclusion in the benchmark, underscoring its legitimacy and ethical soundness.

Table 2: Overview of FLUE Benchmark Datasets

Name	Task	Source	Size (Train/Val/Test)	Metric	License
FPB	Sentiment Classification	[Malo (et al.), 2014]	3488 / 388 / 969	Accuracy	CC BY-SA 3.0
FiQA SA	Sentiment Analysis	[Macedo (et al.), 2018; Shah (et al.), 2022]	822 / 117 / 234	MSE	Public
NHC	News Headlines Classification	[Sinha, Khandait, 2021]	7989 / 1141 / 2282	Avg F1 Score	CC BY-SA 3.0
FinNER	Named Entity Recognition	[Alvarado, Verspoor, Baldwin, 2015]	932 / 232 / 302	F1 Score	CC BY-SA 3.0
FinSBD3	Structure Boundary Detection	[Au, Ait-Azzi, Kang, 2021]	460 / 165 / 131	F1 Score	CC BY-SA 3.0
FiQA QA	Question Answering	[Macedo (et al.), 2018; Shah (et al.), 2022]	5676 / 631 / 333	nDCG, MRR	Public

The emergence of the FLUE benchmark is driven by the necessity to standardize the evaluation of models in the field of financial language understanding. The financial sector imposes unique requirements for processing textual data, such as high terminological complexity, market dynamism, and specific tasks (e.g., sentiment analysis of news headlines, information extraction, etc.). These factors have led to the creation of a heterogeneous set of tasks unified within FLUE, thereby enabling a holistic assessment of different models. Thus, FLUE serves as an essential tool for researchers, facilitating objective model comparison and the identification of areas for further improvement in financial NLP approaches.

1.4 Bridge

1.4.1 Bridge

CHAPTER 2. PRACTICAL SOLUTION

2.1 Limitaions

Prior to practical implementation, a comprehensive analytical evaluation was conducted based on preliminary semantic analysis and structural design to identify and formalize the technical constraints affecting the FinABYSS architecture. As a result, five key problem areas were identified, of which three were able to offer sustainable solutions, while two less critical ones were left for further research.

Deduplication of texts. De-duplication is a necessary element of the text sentiment analysis pipeline because financial signals propagating with delay may be repeatedly fed into the corpus, and their repeated consideration distorts prediction results. For example, the initial publication of mortgage origination irregularities in September 2008 generated strong negative sentiment at the time of the event's occurrence, while its republications years later accompanying retrospective reviews do not have a similar effect on asset prices. Such discreteness of temporal context is not accounted for in classical textual deduplication based on linguistic or syntactic similarity.

Two news items can have almost complete textual similarity and yet carry dramatically different semantic connotations. As a theoretical example, if in the original article about the Citi Group fine (Section 3.2.4), "79 million" had been mistakenly replaced with "79 thousands", it would have had a fundamentally different sentiment and a different meaning.

A special category of duplicate materials is represented by corrections on the Yahoo! Finance platform. Such publications begin with the marker "/CORRECTION/", then the article points to the corrected fragments and repeats the main text almost verbatim. Linear de-duplication at the line or n-gram level in such cases removes the latest version, which violates the logic of stream analysis.

Thus, de-duplication in a financial media stream should fulfill two requirements: first, to distinguish semantically equivalent but contextually different publications in time; second, to correctly handle "correction"-versions, extracting non-overlapping semantics if it differs meaningfully from the previous version. Each document should be considered unique if it contains new information, even if the textual overlap rate is high.

Since the problem lacks source markup, it requires a formal formulation of the semantic deduplication problem that goes beyond simple textual comparison. The semantic space of texts, unlike the lexical space, is continuous and unbounded, and therefore pairwise comparison of strings or texts by cosine distance is not sufficient to check the uniqueness of a single document: it is necessary to operate with volumes of vector representations and to take into account the distribution of ideas in a wider context.

A full description of the mathematical formulation of the problem and the proposed analytical solution is presented in Section 3.3.1. It is there that a strict definition of semantic uniqueness is given, and an algorithm for detecting documents that are close in meaning but differ in information

value is presented.

Computing Resources. Training models on big data requires an extremely powerful computing infrastructure. For example, one FinBERT model was trained on four NVIDIA Tesla P100s for two days on 5 billion tokens [Y. Yang, UY, A. Huang, 2020]. It should be clarified that the training time of only one model is given here, and several more models are usually trained during the experimentation process. Speaking of ModernBERT — it was trained on eight NVIDIA H100s in 10 days, and the total corpus was about 3 trillion tokens [Warner (et al.), 2024].

In the context of our study, ModernBERT is used as the baseline model, so we should focus on it. In the course of the study, a corpus of about 1 billion tokens was collected, which, although smaller than the ModernBERT corpus, would be sufficient for domain-adapted pre-training, i.e., an additional fourth stage of pre-training [Gururangan (et al.), 2020; Warner (et al.), 2024].

Training even the basic ModernBERT model requires extremely powerful computing hardware. The minimum requirements are NVIDIA Tesla T4 from the server segment or NVIDIA RTX 3090 from the customer segment [Warner (et al.), 2024]. However, in the context of the study, only NVIDIA RTX 3060 was available, on which it would have been impossible to adapt ModernBERT. Therefore, due to the difficult availability of computational resources, the context of the study uses a baseline version of ModernBERT fine-tuned for the task of clustering vector representations.

Data. The lack of a representative text corpus in the financial domain was one of the key challenges of this study. Attempts to find open-source datasets on HuggingFace, Kaggle, GitHub and similar platforms revealed that the available financial corpora have either been removed for copyright infringement reasons, are closed for public use, or are too short fragments of text unsuitable for topic modeling and sentiment analysis in the current long-context research field [Daudert, 2022; Macedo (et al.), 2018; Malo (et al.), 2014; Wiebe, Wilson, Cardie, 2005; Xing (et al.), 2020]. In addition, many of the remaining sets are solely for fine-tuning neural network models, but do not contain the necessary amount or variety of data.

Self-collection of financial news articles is complicated by the fact that key sources are disparate and often resist mass scraping. Few providers provide access to paid and expensive APIs (e.g., X, formerly Twitter), where prices for historical data can run into the thousands of dollars. Meanwhile, news sites tend to focus on narrow segments of industry content, making single-source corpus biased and fragmented.

From a financial point of view, not only the breadth of coverage is important, but also the availability of reliable metadata: timestamps, information about the author and others. Many sites lack open archives, provide only RSS feeds, and some resources simply do not place publication time in HTML, which makes automated parsing impossible without in-depth analysis and regular script updates. Different HTML template structures and dynamic content generators (JavaScript rendering) increase the complexity of developing data extraction pipelines.

Thus, trying to assemble a truly representative corpus would require integrating multiple sources with heterogeneous site layouts, resulting in a high technical burden. Focusing on a single

data provider risks introducing an imbalance in thematic and regional representations of financial events. At the same time, existing paid alternatives (e.g., commercial datasets of News from Bright Data[7]) are priced at several thousand dollars for a volume comparable to the corpus collected during the study, which is beyond the budget and research constraints of the project.

Consequently, the lack of open, large and homogeneous financial corpora remains a serious obstacle for building scalable and robust models for topic modeling and tone analysis in finance. The following Section 2.2.2 describes the chosen compromise approach to data collection and aggregation, taking into account the identified limitations and quality requirements of the corpus.

Context window limitation. The base version of the ModernBERT model provides a context window of 8,192 tokens, which significantly outperforms traditional BERT-like models with a 512 token limit [Devlin (et al.), 2019; Warner (et al.), 2024]. However, even with this extended resource, 10-K and 10-Q financial documents, often exceeding tens of pages, do not fit completely. While there are techniques for sliding windowing or fragmenting text into overlapping chunks, they are beyond the scope of the original task of evaluating the out-of-the-box capabilities of modern LLMs. In the context of this study, it was decided to limit the analysis to news articles whose sizes fit within 8,192 tokens, and not to consider multi-page analytical reports. This allowed us to maintain the focus on comparing embodiment and topic models without complicating the preprocessing by aggregating excerpts of long documents.

The irrelevance of existing benchmarks. Comparing the performance of the ModernBERT model and its domain-adapted version on the widely accepted FLUE benchmark seems like a natural step for evaluating progress. However, FLUE datasets consist mostly of short chunks (up to 512 tokens) designed for typical text comprehension tasks [Shah (et al.), 2022]. Because these datasets are sharpened for the 512-token window, they do not reflect the benefits of ModernBERT's extended context and, conversely, will underestimate its totals. Thus, the use of FLUE in the current study will lead to a distorted perception of the model's qualities: short-text tasks do not demonstrate its ability to capture long-term dependencies and synthesize information from large amounts of data.

Both problems — the limited context window when analyzing long documents and the unrepresentativeness of standard 512-token benchmarks — dictate the need for specialized preprocessing and validation techniques tailored to the financial domain. With the exception of these two problems, all other problems have been solved and their solutions are proposed in further sections of the paper.

2.2 Data Governance

2.2.1 Data Requirements

One of the study's key objectives was data collection. As noted, the corpus was constructed to meet requirements for universality and to support future research in related fields.

[7]URL: `https://brightdata.com/products/datasets/news`

Data Requirements. Texts were chosen as the primary source of qualitative data. In the financial domain, the most content-rich and impactful on asset prices are:

- news articles;
- social-media posts;
- official reports (annual, quarterly, strategic);
- press releases;
- analytical reviews and articles;
- transcripts of interviews, conferences, and public-company webcasts.

Previous studies have validated the effectiveness of these text types. FinBERT [A. Huang, H. Wang, Y. Yang, 2023; Y. Yang, UY, A. Huang, 2020], for example, was trained on official reports and analytical articles, and its later versions incorporated press releases [Liu (et al.), 2020]. Other models have been successfully pre-trained on fragmented news articles and fine-tuned for sentiment analysis on news headlines and social-media publications [Araci, 2019]. Nevertheless, our research deliberately processes full texts without fragmentation: official reports frequently exceed the 8,192 token limit of ModernBERT, complicating their integration, while very short formats (social-media posts) fail to exploit the advantages of a long-context model.

Furthermore, no single source aggregates all of the above text types. Given limited resources, the following most significant categories were selected for initial focus:

- news articles;
- press releases;
- analytical reviews and articles;
- transcripts of financial events.

Expansion of the corpus to include additional content categories is planned for future work.

To ensure the corpus's versatility and facilitate subsequent use, the following metadata were collected for each text:

- Headline (e.g., "Covestro board enters formal talks on $12 billion ADNOC approach");
- Source (copyright holder), e.g., Reuters, Simply Wall St., PR Newswire, @ilyasut, Max Gottich;
- Publication platform, e.g., Twitter, Yahoo! Finance, Reddit, Seeking Alpha;

- UTC timestamp with second-level precision (e.g., 2024-09-01T01:48:13);
- Author-assigned topical tags (e.g., [“M&A”, “Cryptocurrency”, “Tech”])’
- List of tickers mentioned by the author (e.g., [“9626.HK”, “BILI”]).

Finally, the optimal period for data collection was established. The lower and upper bounds of the dates were selected based on the assumption that this corpus will be used to train the value prediction model in the future, which requires that the initial knowledge of ModernBERT be synchronized with those on which it will be fine-tuned later. Due to the fact that the ModernBERT publication does not disclose the data on which the model was trained [Warner (et al.), 2024], it is impossible to accurately judge for which period the data was taken. Therefore, in our study, focusing on the date of ModernBERT publication [Ibid.] and the classical ratio of training, validation and test samples as 75/15/10, we took the date range from September 17, 2023 to March 18, 2025, i.e. 548 days, of which 374 are working days according to the US calendar.

Source Requirements. Data sources must be open, free, English-language, and authoritative, since wide dissemination and timely publication directly affect market reactions. Considered sources included:

- News outlets: Bloomberg[8], The New York Times[9], Reuters[10], etc;
- Analytical platforms: Seeking Alpha[11], TradingView[12];
- Official sites: Corporate and government portals.

An analysis of over 50 corporate and more than 100 government resources showed that, thanks to RSS feeds, press releases are centrally aggregated via PR Newswire[13] and GlobeNewswire[14]. Other automated aggregators (e.g., Business Wire[15]) exist, but PR Newswire and GlobeNewswire empirically dominate; however, they offer press releases only, without genre diversity.

Among news outlets, Reuters proved optimal in responsiveness and market coverage. Niche but high-quality sources (e.g., The Information[16], Epoch AI[17]) were also examined; they provide overly specialized content, whereas traditional outlets such as Bloomberg, The Wall Street Journal[18], and The Economist[19] cover a broader market spectrum, albeit with technical constraints.

[8]URL: `https://www.bloomberg.com/`
[9]URL: `https://www.nytimes.com/`
[10]URL: `https://www.reuters.com/`
[11]URL: `https://seekingalpha.com/`
[12]URL: `https://tradingview.com/`
[13]URL: `https://www.cision.com/`
[14]URL: `https://www.globenewswire.com/`
[15]URL: `https://www.businesswire.com/`
[16]URL: `https://www.theinformation.com/`
[17]URL: `https://epoch.ai/`
[18]URL: `https://www.wsj.com/`
[19]URL: `https://www.economist.com/`

Of the analytical platforms, Seeking Alpha was excluded due to its large volume of paywalled content, and TradingView was unsuitable because it does not grant access to historical publications.

Thus, PR Newswire and Reuters became the corpus's primary sources. To mitigate the risk of systematic bias and ensure broader coverage, it was nevertheless decided to enrich the corpus with additional publishers. Due to their fragmentation, lack of APIs, and often incomplete metadata (including sub-second timestamps), aggregation and synchronization proved unfeasible. Consequently, the aggregators Google Finance[20], Yahoo! Finance[21], and FinURLs[22] were also considered.

Ultimately, Yahoo! Finance was selected thanks to its unified site structure and comprehensive aggregation of diverse sources, whereas FinURLs redirects to individual source sites, each with its own layout. Google Finance, though similar to Yahoo! Finance, does not support the collection of historical data.

2.2.2 Data Collecting

Prior to data collection, a survey and analysis of existing open-source tools for harvesting data from Yahoo! Finance were conducted. The analysis identified five candidate libraries. Two — yahooquery and yahoo-stock-api — do not support article extraction; two others — yahoo_fin and fin-news — are abandoned and no longer function correctly; and yfinance affords access only to the latest twenty news items in real time. Consequently, a custom Python parser was developed. Its architecture comprises two principal stages:

1. Link Collection. A recursive traversal of the official sitemap is used to gather article URLs for a specified period. Each "daily page" lists 50 news links and a pointer to the next page; critically, page n can only be accessed via page $n-1$, creating a bottleneck akin to traversing a linked list under high network latency.

2. Content Extraction. The gathered URLs are then parsed to extract each article's text. It should be noted that, as with training the original BERT model, tables and images are not processed [Devlin (et al.), 2019].

During development, several constraints were encountered and subsequently addressed in the parser's design:

- IP-blocking and Cookies. Yahoo! Finance limits to 14 concurrent requests per IP at minimum 4-second intervals; violations yield HTTP 404, 429, or 200 responses with empty bodies. Even when these constraints are met, blocks may still occur. To mitigate this, a

[20]URL: `https://www.google.com/finance/`
[21]URL: `https://finance.yahoo.com/`
[22]URL: `https://finurls.com/`

pool of 50 proxy servers was employed, and failed requests were automatically retried in subsequent iterations.

- Regional Restrictions. Identical URLs may be inaccessible or behave inconsistently when requested from different countries.
- Technical Errors. Redirects to external sources, broken links, and paywalled URLs were encountered and excluded during corpus assembly.

To accelerate processing of large datasets, the C-based library selectolax was used, offering roughly 30× the speed of BeautifulSoup and 5× that of lxml.

As a result, the link-collection stage yielded 1,362,103 URLs, of which 1,360,761 belonged to the Yahoo! Finance domain. Thanks to the parser's modular architecture, extensive proxy usage, and multiple iterations, 1,304,717 articles were successfully parsed. The final corpus occupies 6.5 GB in CSV format and 2.2 GB in the more compact Parquet format.

The final class implementing the parser program can be found in the official repository of the project, called YahooFinanceParser[23].

2.2.3 Data Analysis

Before commencing the pre-processing stage, it was decided to conduct a comprehensive analysis of the collected corpus of news articles. This preliminary analysis not only revealed the characteristic features of the data but also established the foundation for subsequent automation of text cleaning and structuring. Moreover, the analysis results have also impacted the quality of the trained model.

Local analysis encompassed a detailed examination of various subsets of the corpus aimed at identifying patterns characteristic of non-representative or "noisy" articles. In this process, key signals—such as specific keywords in the titles and opening paragraphs—were identified that allow for the automatic filtering out of undesirable texts. Furthermore, the local investigation uncovered potential rules for removing marketing fragments, metadata, and other artifacts that adversely affect data quality. All the obtained rules were subsequently formalized (see Section 2.2.4 for further details).

Global analysis is dedicated to studying the central tendencies of the corpus through descriptive statistics and the analysis of various data representations—both metadata and the textual content itself. This approach enabled the evaluation of the distribution of key characteristics, the identification of seasonal and thematic patterns, and the preparation of aggregated results that serve as the basis for further refinement of the pre-processing methodology.

Below are the aggregated results of the global analysis, which, together with the local findings, allow for a deeper understanding of the nature of the collected dataset and help determine directions for its further optimization.

[23]URL: `https://github.com/denisalpino/FinABYSS`

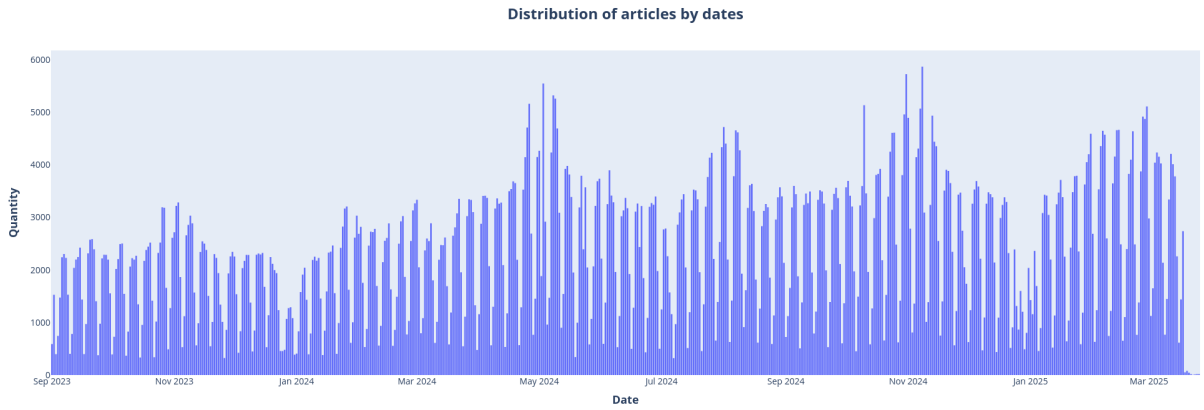

Figure 1: Distribution of publications by dates.

Distribution of publications by dates. Figure 1 shows that the number of publications fluctuates daily with a certain periodicity. A detailed examination revealed that the minimums occur on Sundays and public holidays, when fewer financial news items are published. This naturally reflects the market's characteristics: on weekends and holidays, business activity declines, leading to fewer publications.

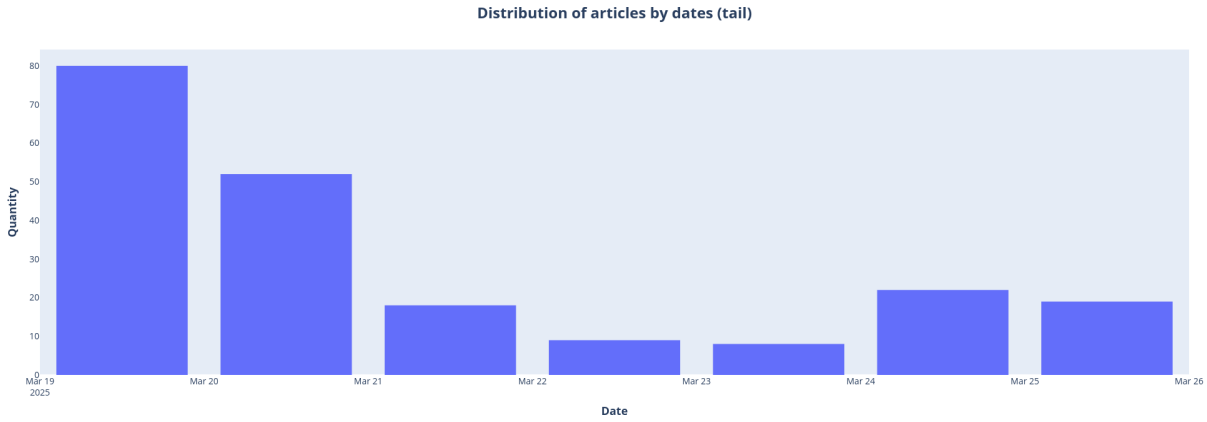

Figure 2: Distribution of publications by dates (tail).

At the same time, some articles, by formal characteristics, fall outside the collection period (September 17, 2023 - March 18, 2025). Figure 2 displays these "tail" publications, whose count slightly exceeds 200. A more detailed analysis determined that these articles were indeed published within the specified interval, but their content was later edited or supplemented. As a result, the publication date and time on the corresponding website were updated, and the old version (with the original date) was lost. Had the links been parsed not after one week but several weeks later, more such cases would have been observed.

From the perspective of short-term market forecasting, this circumstance may lead to distorted timestamps, making some articles appear to have been published later than they actually were. Therefore, the dataset might prove less effective for short-term studies compared to medium-

and long-term ones (where a shift of a couple of days is less critical). Nevertheless, for this work it does not play a crucial role, as the model relies solely on the text of the article and does not take into account the precise publication timestamps.

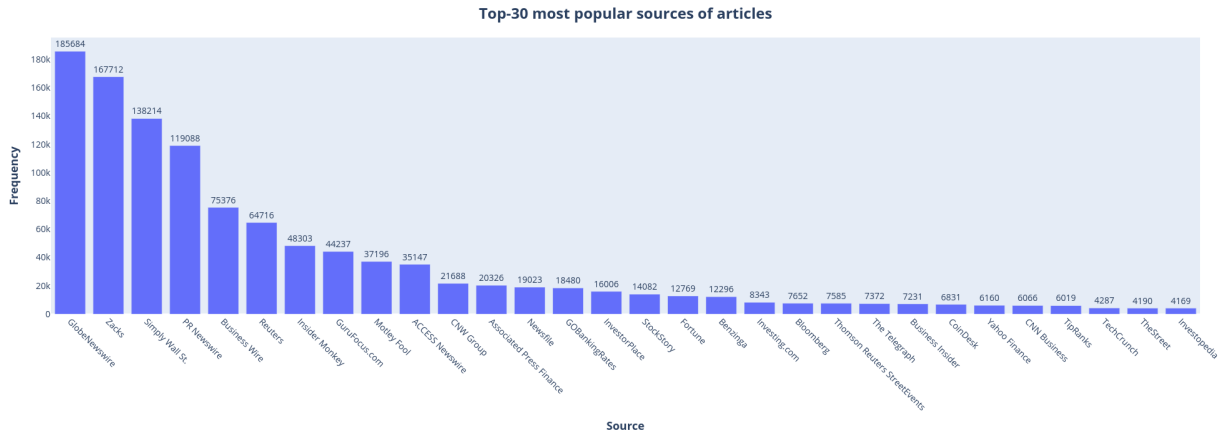

Figure 3: Top-30 most popular sources of publications.

News Sources. Figure 3 illustrates the distribution of publications across the 30 most frequent sources. The analysis showed that the predominant share of articles (potentially 69.2%) was published by semi-automated aggregators: GlobeNewswire, Zacks, Simply Wall St., PR Newswire, Business Wire, GuruFocus.com, Motley Fool, among others. These aggregators focus on the automatic collection of key data from various resources (regulators, official company websites, etc.), publishing press releases, brief report summaries, and invitations to corporate events.

Among the top 15 sources, only some can be conditionally considered as "traditional" news outlets, such as Reuters, Insider Monkey, CNW Group, Associated Press Finance, and InvestorPlace. Meanwhile, outside the top 30, classic publications that primarily publish original articles prevail. In reality, the blurred boundary between original and semi-automatically generated content complicates efforts to clearly differentiate them.

According to approximate estimates, out of 1 300 000 articles, about 900 000 (69.2%) are semi-automated. This is an important factor for training a language model because:

1. The quality of such materials is often lower: texts contain artifacts, broken formatting, and incorrectly inserted characters.

2. Their volume is large, which, on one hand, provides a substantial sampling capacity, but on the other, complicates cleaning and normalization without the loss of significant information.

Nevertheless, even "imperfect" texts from aggregators convey useful information about the financial market and companies. However, it is extremely important to develop appropriate cleaning and pre-processing rules (discussed in detail in Section 2.2.4) to preserve the semantic integrity of the texts.

Furthermore, and perhaps more importantly, these semi-automated texts contribute roughly the same total number of tokens as the "original" articles (30.8%), despite their numerical dominance. Consequently, with proper processing, this group of semi-automated articles can make a significant contribution to training the language model without diminishing the value of the original texts.

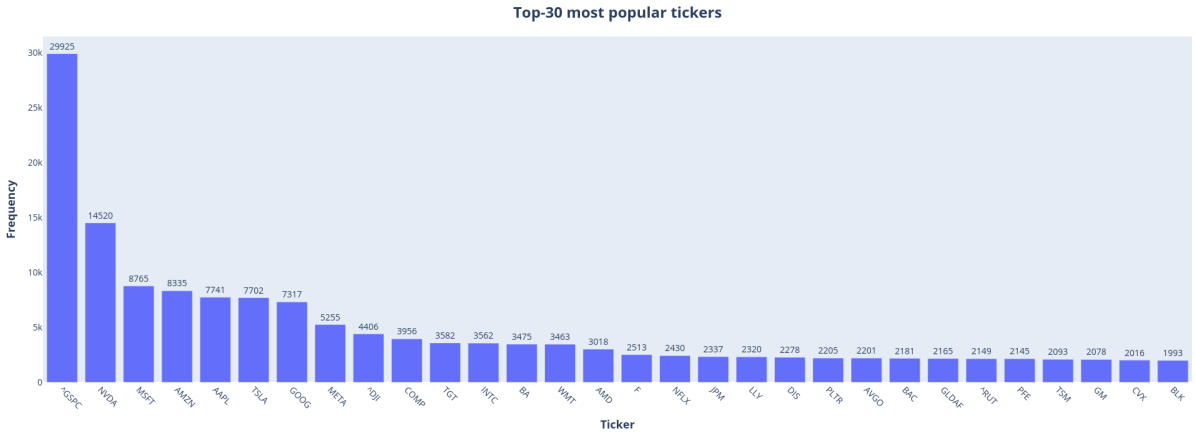

Figure 4: Top-30 most popular tickers.

Analysis of tickers. Figure 4 presents the distribution of publications by the 30 most frequently mentioned tickers. The leader is the S&P 500 index, although the sample also includes the Dow Jones and Russell 2000. Notably, the top 10 are predominantly IT companies, with Nvidia leading by a significant margin.

At the same time, approximately 574 000 (44.2%) publications do not contain any tickers in the article header. Moreover, even when tickers are present, they may not reflect all the companies or indices mentioned in the article. This indicates that although this dataset column is fairly representative, it does not provide complete coverage of all potential tickers, and some news items are formally omitted from consideration. Therefore, for tasks beyond the scope of this research, it would be advisable to create a dictionary of terms and names associated with each specific ticker and then algorithmically augment the ticker column using the corresponding texts.

Figure 5: Wordcloud of the whole collected corpus.

Text Quality. Figure 5 shows a word cloud generated from the entire corpus of collected texts. From this visualization, the following key conclusions can be drawn:

1. **Data Representativeness**. The word cloud demonstrates a wide range of financial terms, indicating that the dataset is sufficiently representative of financial topics. This suggests that the material covers various aspects of market activity and economic events.

2. **Specificity of Financial Terminology**. The frequency distribution of financial terms significantly differs from that observed in popular corpora used for training language models (e.g., English Wikipedia or BookCorpus). This discrepancy necessitates the application of DAPT to effectively train the model on domain-specific financial data.

3. **Level of Noise and Presence of Irrelevant Information**. The word cloud includes elements such as “Zacks”, “click”, “please”, “free”, and “source”. This indicates a significant presence of noisy, promotional, or automatically generated fragments, which calls for the development of specialized methods for data cleaning without compromising the semantic integrity of the texts.

Additionally, it can be noted that the identified noise and dispersion of terms may negatively affect the quality of downstream tasks, such as classification or embedding extraction, if the data is not properly processed during the pre-processing stage.

Summary. The collected news dataset is characterized by several notable features. Firstly, there is pronounced seasonality in the publications—the minimums occur on weekends and public holidays, and so-called "tail" articles have also been recorded. Secondly, the analysis of sources

indicates that about 69% of the texts originate from semi-automated aggregators, which can complicate the data cleaning process, as such sources often yield texts with broken formatting, embedded artifacts, and irrelevant information. Finally, it has been determined that the dataset exhibits a high variability of financial terminology while also containing a significant level of noise, which altogether confirms the need for DAPT and the development of effective text cleaning methods.

On one hand, the identified features (time shifts, noise, dominance of semi-automated sources) may reduce the suitability of the dataset for short-term forecasting or tasks that require precise timestamping. On the other hand, for tasks oriented toward the semantic content of the text, these issues do not have a critical impact. Proper pre-processing, including text cleaning and the removal of irrelevant elements, will substantially improve the quality of the trained model and expand its ability to generalize across various types of publications.

2.2.4 Data Preprocessing

After local and global analyses of the text data, which revealed both noise patterns within individual documents and system anomalies indicating the unrepresentativeness of some publications in the corpus as a whole, a multi-stage streaming on-the-fly preprocessing pipeline was designed using chunks of 100,000 publications. A key limitation in the development of the solution was the limited memory: with the original corpus size of about 15 GB, standard filtering and full text traversal operations required the allocation of a significant constant memory buffer. This necessitated adapting the algorithm to stream processing in fixed-size chunks.

During the first stage, each chunk was downloaded from the repository and immediately subjected to initial filtering: documents containing less than 100 characters were rejected as insufficiently informative. The empirically set threshold of 100 characters turned out to be sufficient to eliminate “empty” artifacts arising due to markup instability on the source side. For example, on popular sites like Yahoo! Finance, text is sometimes stitched inside atypical HTML tags — for example, in ’<tbody>’ instead of the familiar ’<p>’ — resulting in almost entirely whitespace entries or marketing inserts at the end of the document.

The next step was to implement a headline filter: 66 rules were formalized based on it, covering both generic patterns (’Form 8’, ’Net Asset Value’, ’Holdings in company’) and more targeted for noisy 12 sources (for example, GlobeNewswire removed headlines containing ’Declaration’ or starting with ’Key digital’). This selection ensured the removal of publications consisting predominantly of tabular data or short fillers.

After normalizing all kinds of whitespace - merging consecutive spaces or line breaks into a single character, replacing unbroken spaces ’
xa0’ with a standard space, and unifying special characters in the text — 12 more rules were applied to reject unrepresentative documents based on the content of the main body of the publication. In particular, anything starting with the “(Repeat)” marker, as well as welcome templates of specific sources like “Dear madam, sir, please find hereunder the links” for GlobeNewswire, were

automatically excluded from further processing.

Next, the text was cleaned of contact information and typical footers: links, email addresses, and phrases that clearly indicate the end of the document (“Forward-looking statements”, “Contact Details”) or the unrepresentativeness of a paragraph (“Source:”, “See More:”, “Sponsored:”). In addition, standard introductory phrases such as “The recommendations of Wall Street analysts” and “When deciding whether to buy, sell, or hold a stock, investors often rely on analyst recommendations” were removed from the beginning of publications. A total of 92 cleaning rules were generated across the different groups.

This aggressive cleaning of texts is due to the fact that filtered patterns are so frequent that they have an extremely negative impact on clustering, artificially inflating the clustering metric. After the aggressive cleaning of the corpus, the quality metric did decrease, but the clusters became much more representative and based on the semantics of the news itself rather than its sources, marketing and legal information in the texts.

Figure 6: Wordcloud of the whole collected corpus after text and documents preprocessing.

As a validation of the quality of text cleaning, we can refer to the word cloud of the corpus after preprocessing (Figure 6), which clearly shows the improvement: words such as ’forward-looking’, ’link’, ’click’, ’simply’ (source name Simply Wall St.) and others are missing.

Finally, before saving the chunks, sorting by publication date and deleting duplicates based on the full text of the document took place. Taking into account the described operations, the size of the corpus was reduced from 15.4 GB in CSV format (8.6 GB in Parquet) to 5.9 GB (2.0 GB in Parquet): out of 1,304,717 original records, 1,267,416 remained, i.e., only 2.8% of the documents were deleted. This result indicates a high proportion of text noise in the collected dataset and emphasizes the effectiveness of the multi-stage stream-based preprocessing approach,

which saves RAM and provides high-quality cleaning of text data before subsequent stages of thematic modeling and analysis.

2.3 Model Development

2.3.1 Feature Extraction

After preprocessing the text corpus and removing background noise, embeddings were extracted to accelerate subsequent stages of topic modeling, including dimensionality reduction and clustering.

The base ModernBERT model is not optimal for this task, since its vector representations are excessively sparse. High sparsity of embeddings degrades clustering quality, particularly for density-based methods (e.g., DBSCAN). Although HDBSCAN is more robust to density variations, sparsity still adversely affects clustering outcomes.

To address this issue, it is common to fine-tune the model on a semantic textual similarity (STS) task. In this setting, the model receives a pair of texts and returns a similarity score [Muennighoff (et al.), 2023], typically computed via cosine distance (Formulation 4).

$$D_{\cos}(\mathbf{u}, \mathbf{v}) = 1 - \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|_2 \, \|\mathbf{v}\|_2}. \tag{4}$$

STS models consistently produce denser and more informative embeddings.

Accordingly, for the base ModernBERT we evaluated two fine-tuned variants: modernbert-embed from Nomic AI [Nussbaum (et al.), 2024] and gte-modernbert-base from Alibaba [Z. Li (et al.), 2023; X. Zhang (et al.), 2024]. Evaluation employed the Massive Text Embedding Benchmark (MTEB), which spans eight task categories and 58 datasets [Muennighoff (et al.), 2023]. Results were as follows:

- Clustering (12 datasets): gte-modernbert-base outperformed modernbert-embed by 1.5 percentage points (44.98% vs. 44.47%).
- STS (10 datasets): Their performances were comparable (81.78% vs. 81.57%).
- Overall (eight tasks, 56 datasets): gte-modernbert-base led by 1.76 percentage points on average (64.38% vs. 62.62%).

Consequently, gte-modernbert-base from the sentence_transformers library was chosen for embedding extraction [Reimers, Gurevych, 2019]. Instead of using the [CLS] token, a more advanced Mean Pooling technique was applied, averaging token embeddings across the sequence (Formulation 5).

$$\mathbf{h}_{\mathbf{mean}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{h}_t. \tag{5}$$

where T is the tokenized sequence length and $\mathbf{h}_t$ denotes the embedding of the t-th token.

To accelerate the training of dimensionality reduction and clustering models, the embeddings were first reduced to the unit L_2-norm (Formulation 6):

$$\widehat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|_2}, \qquad \|\mathbf{u}\|_2 = \sqrt{\sum_{i=1}^{n} u_i^2}. \tag{6}$$

Such preprocessing allows us to use the GPU-optimized Euclidean distance metric (Equation 7) when training the dimensionality reduction model, without having to repeat the L_2-norm computation that occurs when computing the cosine distance metric at each iteration of the hyperparameter optimization. so that Euclidean distance could be used for both dimensionality reduction and clustering.

$$D_2(\mathbf{u},\mathbf{v}) = \|\mathbf{u}-\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^{n} \left(u_i - v_i\right)^2}. \tag{7}$$

Thus, after computing the L_2-norm and the Euclidean distance, we can actually, in a sense, be considered to be working with the cosine distance, since the reduced measure becomes monotonically related to the cosine and reflects the same order of proximity of the points (Equation 8), but all computations are accelerated by GPU optimization.

$$D_2\left(\widehat{\mathbf{u}},\widehat{\mathbf{v}}\right) = \left\|\widehat{\mathbf{u}}-\widehat{\mathbf{v}}\right\|_2 = \sqrt{2\left(1-\widehat{\mathbf{u}}\cdot\widehat{\mathbf{v}}\right)}. \tag{8}$$

To build and train the dimensionality reduction and clustering algorithms, a training subsample of 200,000 embeddings and associated metadata was generated, which is approximately 16% of the entire corpus. This size of the training subsample was chosen based on the available computational resources. Thus, 200,000 embeddings were used to select the optimal hyperparameters, while the remaining 1,050,000 embeddings were reserved for the validation and inference phases. At the same time, before inference, the pipeline of dimensionality reduction and clustering models were trained on the entire corpus with a linear increase in hyperparameter values for the HDBSCAN algorithm, the choice of which is conditioned in Section 2.3.2.

Moreover, mixed precision (float16) and the FlashAttention mechanism [Dao (et al.), 2022] were employed during embedding extraction, substantially reducing computational resource requirements and runtime.

2.3.2 UMAP and HDBSCAN

Thus, after assembling the embedding sample, we proceeded to experiments with dimensionality reduction and clustering models, performing their joint optimization. This approach reflects the multi-criteria nature of the task: it is necessary not only to preserve the structural (global) and local relationships from the original 768-dimensional space, but also to ensure that embeddings

remain separable ("clusterable") in a low-dimensional projection suitable for two-dimensional visualization.

As key requirements we identified:

1. Preservation of cluster structure. Embeddings after dimensionality reduction must remain separable, preserving groupings by semantic and topical similarity.

2. Suitability for two-dimensional visualization. The resulting space must support clear and interpretable planar display.

To search simultaneously for optimal hyperparameters of both dimensionality reduction algorithms and clustering methods, we employed a unified meta-optimization process.

We used the DBCV index as our optimization metric, since it does not assume any predefined cluster shape (unlike as example silhouette coefficient favoring spherical or ellipsoidal structures) and effectively evaluates density-based clustering methods.

As our base clustering algorithm we selected HDBSCAN [Campello, Moulavi, Sander, 2013], which meets two crucial requirements:

- No shape assumptions. Unlike K-Means, HDBSCAN does not assume clusters are Gaussian spheres, which is critical for representing topics.

- Hierarchical, density-based nature. It can identify both large thematic groups and small, highly concentrated niches.

Moreover, GPU acceleration of HDBSCAN yielded high processing speed on both large samples and high-dimensional data.

Within the HDBSCAN framework, we tuned two key hyperparameters: the minimum number of neighbors — the count of points in a neighborhood required to consider a point a cluster "core" — and the minimum cluster size — the threshold number of observations for forming a cluster, which allows capturing rare, narrowly topical groups.

Pilot experiments revealed the coexistence of very dense regions ("hot topics") and rare but semantically significant clusters. A small minimum cluster size captures these rare topics but also increases the number of micro-clusters, some of which lack clear semantic distinction.

To mitigate this, we considered an ε-based cluster-merging technique [Malzer, Baum, 2020], consolidating adjacent micro-clusters in high-density regions. However, this approach complicates inference: when new observations arrive, ε-merging cannot be incrementally updated, necessitating full retraining.

Prioritizing practicality, we therefore abandoned ε-merging in favor of smaller minimum cluster sizes, accepting some fragmentation while preserving the ability to interpret and agglomerate clusters at higher hierarchical levels.

Another hyperparameter — the cluster selection method — determines whether clusters form based on excess of mass or tree leaves. We found that the latter yields finer-grained, more homogeneous groups, and used it for our final configuration.

Thus, in the final optimization stage, only two HDBSCAN parameters remained tunable: minimum neighbors and minimum cluster size.

It is noteworthy that, as will be described in Section 3.3, the future architecture assumes a fixed number of clusters corresponding to a static number of experts; hence, we employ the cuML implementation of HDBSCAN rather than its adaptive variant [Vijayan, Aziz, 2022].

Turning to dimensionality reduction algorithms, our preliminary selection included t-SNE, PCA, UMAP, TriMap, and PaCMAP, with key evaluation criteria of fidelity, the ability to balance local and global relationships, and training time complexity. Based on other researchers' experience, UMAP was chosen as the baseline algorithm [Grootendorst, 2022].

t-SNE was excluded due to its insufficient scalability on large datasets. PCA, although fast as a global approximation, did not preserve local structure in the final low-dimensional embedding, and an experiment combining PCA with UMAP fell short of standalone UMAP by approximately 37% in DBCV score. While TriMap and PaCMAP achieved similar performance in intermediate dimensions — and PaCMAP produced a more uniform distribution for two-dimensional visualization — the GPU-accelerated implementation and demonstrated robustness of UMAP in cuML led us to select it as the definitive method for both intermediate dimensionality reduction and final 2D projection.

2.3.3 Hyperparameters Optimization

The model initialization phase was followed by a comprehensive hyperparameter optimization involving seven key parameters: five for UMAP (number of neighbors 'n_neighbors', output space size 'n_components', minimum distance between points 'min_dist', parameter 'spread' and coefficient 'negative_sample_rate') and two for HDBSCAN ('min_cluster_size' and 'min_samples'). Ray Tune [Liaw (et al.), 2018] and Optuna [Akiba (et al.), 2019] were tested as frameworks for finding the optimum, with an emphasis on a resource-based approach where the 'resource' was the size of the training subsample. Experimentally, we selected 5 training stages with shares from 1.5% to 10% of the total corpus, which allowed us to evaluate the scalability of the methods and the stability of the obtained hyperparameters.

In the context of Ray Tune, the BOHB (Bayesian Optimization with HyperBand) scheme [Bergstra (et al.), 2011; Falkner, Klein, Hutter, 2018; L. Li (et al.), 2018; Shahriari (et al.), 2015] was used, but it did not provide a proper speed-to-quality ratio when working with UMAP/HDBSCAN. Switching to Optuna, three varieties of "pruners" — mechanisms for early cutoff of unpromising combinations during resource build-up — were implemented. The first, 'AdaptiveStablePercentilePruner', removed the worst ones by a predefined percentile in each resource step, the second, 'CustomPatientPruner', minimized the risk of premature exclusion by

cutting off combinations after n consecutive unsuccessful steps with gains less than δ, and the third, 'NormalPruner', used Z-statistics (Equation 9) and a given percentile of the standard normal distribution ($\mathcal{N}(0,1)$).

$$z\text{-score} = \frac{m - \mathbb{E}[M]}{\mathbb{V}[M]}, \tag{9}$$

where m is the current metric and M is the distribution of metrics at this stage.

Weighted Cumulative DBCV-index (WCDBCV) became the evaluation metric during the iterations:

$$WCDBCV_j = \frac{1}{j\sum_{i=1}^{j} p_i} \sum_{i=1}^{j} p_i \cdot DBCV_i, \tag{10}$$

where j is the current resource step and p_i is the fraction of the sample used at the i-th step.

Of all combinations, the best performance was demonstrated by the Tree-structured Parzen Estimator (TPE) algorithm [Bergstra (et al.), 2011; Shahriari (et al.), 2015; Watanabe, 2023] in conjunction with 'AdaptiveStablePercentilePruner', but the quality gain was not significant enough relative to the increased computational cost. Therefore, the subsequent hyperparameter search was performed without the use of pruning. As a global strategy, a classical TPE with 400 trails was used, half of which were performed in a "warm-up" mode of random configurations. An Covariance Matrix Adaptation Evolution Strategy (CMAES) [Auger, Hansen, 2005; Nomura, Shibata, 2024] with an IPOP restart mechanism, adaptive learning rate and a preliminary "warm run" based on the 15 best combinations found in the global optimization phase [Auger, Hansen, 2005; Nomura, Akimoto, Ono, 2023; Nomura, Shibata, 2024; Nomura, Watanabe, (et al.), 2021] was used to locally fine-tune the emerged promising regions of the hyperparameter space. The budget for the local phase was 200 trails, which provided a deep search around already high-performing configurations.

As a result of this two-tiered approach — first a broad TPE study, then an in-depth local CMAES search — it was possible to balance the breadth and depth of optimization, improving the robustness of the model pipeline.

2.4 System Development

After completing the semantic clustering step and constructing topic groups, it is necessary to process the generated clusters at the linguistic level. In topic modeling, the quality of topic representations is key to interpreting topics, communicating results, and understanding patterns. Thus, it is necessary to validate that each document set is indeed characterized by a unique set of key terms and does not contain mis-segmentation artifacts. This validation goes beyond purely metric evaluations and requires an empirical, "human" view of which words truly define topic content. In the context of an unlabeled corpus, such validation is particularly important: without initial topic labels, the only source of information about the semantic homogeneity of clusters

remains the text of the publication itself. Moreover, there is a need to assign names to each of the clusters, so it is crucial to develop a topic representation pipeline.

On the other hand, clustering results in lower level labels, and in the context of the current system a hierarchical approach to topic representation is implied, which leads to the need to extract additional information from the current model and post-process it.

To simplify and formalize the process, FinABYSS implemented a specialized topic representation pipeline based on the following tools:

- BERTopic — for convenient and simplified processing of linguistic features, visualization of weighted word frequencies, and hierarchy construction.
- OpenAI API — to automate the partitioning of the obtained clusters based on their linguistic features.
- DataMapPlot — to simplify work with the graphical interface and create an interactive semantic map.
- Proprietary implementation:
  - over BERTopic — to realize the possibility of selecting a certain number of hierarchical thematic levels.
  - between BERTopic and OpenAI — to realize the possibility of assigning thematic names based on linguistic features at higher hierarchical levels than only at the lowest one.
  - between BERTopic and DataMapPlot — for fine-tuning both static and interactive semantic maps.

Thus, the first step in the pipeline was to vectorize the texts, i.e. to bring them to matrix form. For this purpose, the 'CountVectorizer' from the 'sklearn' library was used, which was configured to extract unigrams and bigrams. This is useful for a more accurate representation of topics as it leads to terms such as "central bank", "monetary policy" and "New York" being considered both together rather than separately word by word.

Henceforth, words in the context of documents such uni- and bigrams will be called "terms" to prevent confusion, since in fact, in the context of the developed system, not words but n-grams, i.e., short sequences of n words, are considered.

Also, based on the free English-language stop-word dictionary [Nothman, Qin, Yurchak, 2018], the stage of filtering out articles, prepositions, pronouns, conjunctions and other parts of speech that interfere with the selection of truly representative terms was set up.

Finally, the minimum frequency of a term in a document was configured for its inclusion in the matrix. It is not difficult to imagine a situation where a certain term occurs only once in all

documents. It is unlikely that the term reflects a particular topic. On the other hand, the corpus contains over a million documents and unless a minimum frequency of a term is set, the matrix will become huge and impractical to use. Therefore, terms appearing less than 15 times in the entire corpus are cut off as statistically unrepresentative.

Finally, the system implies real-time operation, so it is fundamental to be able to incrementally update the term matrix, for this purpose an online version of 'CountVectorizer' — 'OnlineCountVectorizer' was used.

After forming the matrix of terms, it is necessary to group them into thematic clusters and determine the relevance of terms for each thematic group, in order to further generate cluster names based on the most representative terms.

Term Frequency Inverse Document Frequency (TF-IDF) and Best Match (BM-25) are additive relevance functions. They are used by most search engines as basic relevance metrics. Both metrics indicate the relevance of a document. The higher the value of the metrics, the more relevant the document is. Having said that, it is important to note that the value of the metrics itself does not have any meaningful interpretation other than the relative difference of relevance of documents or terms with each other.

In the context of search engines, the metric reflects the relevance of a document or term to a search query, but the current system uses a modified TF-IDF metric that expresses the relevance of a term to a topic rather than to a query.

Such a metric, is called c-TF-IDF [Grootendorst, 2022]. It can best be explained as a TF-IDF formula adopted for the topic as a whole. That is, all documents within a topic are not considered separately, but are combined into one large document, on the basis of which the calculation is made. Thus, c-TF-IDF takes into account what distinguishes documents in one topic cluster from documents in another.

Thus, we first extract the frequency of term x in topic cluster c to which term x belongs. This results in a class-based representation of tf. And in order to account for the differences in the size of topic clusters, the L_1-norm is computed from the extracted frequencies.

Then, the logarithm of the sum of the unit and the quotient of dividing the average number of words in each topic cluster by the frequency of word x in all clusters is calculated. Thus, an idf representation is obtained, which helps to determine how rare a given word x is among all other classes. The unit is needed to ensure that the logarithm is always positive.

Finally, as in the case of regular TF-IDF, the obtained representations tf and idf are multiplied (Equation 11):

$$w_{x,c} = ||tf_{x,c}||_1 \times \log\left(1 + \frac{\frac{1}{||\mathbb{C}||}\sum_{i\in\mathbb{C}}||\mathbb{X}_i||}{\sum_{i\in\mathbb{C}} x}\right), \tag{11}$$

where $w_{x,c}$ is the relevance of term x to topic cluster c, $\mathbb{X}_i$ is the set of all terms in topic cluster i.

However, instead of the classical c-TF-IDF formula, it was decided to use its modified analog. Some words or terms occur too frequently in each topic, but are not considered typical stop words for exclusion from the text. To smooth out the most frequent terms in a topic, the system applies term frequency normalization, i.e., it extracts the square root of tf, after applying a weighting scheme.

On the other hand, although the collected financial corpus is large, it may not fully represent all the lexical abundance of the general population. Therefore, for a more stable result, BM-25 transformation was applied to idf. The final relevance formula looks as follows:

$$w_{x,c} = \sqrt{||tf_{x,c}||_1} \times \log\left(1 + \frac{\frac{1}{||\mathbb{C}||}\sum_{i\in\mathbb{C}}||\mathbb{X}_i|| - \sum_{i\in\mathbb{C}}x + 0.5}{\sum_{i\in\mathbb{C}}x + 0.5}\right), \tag{12}$$

The 0.5 coefficients are based on fitting to a theoretically cleaner form. This form gives less weight to terms that occur too frequently.

As a result, we obtain bag-of-words for each of the topic clusters that are fairly representative of the lexical richness of each topic. However, the bag-of-words may still contain some words that are not fully representative or the bag may be overly homogeneous. These problems can negatively affect the topic name generation process, so a more sophisticated pipelining was envisioned in the system design, which adds 2 more steps before the final representation.

The first step is to semantically compare the most representative words, with the most representative documents. First, the 30 most representative terms are extracted from the bag-of-words, and the 5-10 most representative documents are extracted from the topic cluster itself. Then, we collect the already extracted embeddings from the corresponding documents, and using the same embedding model — 'gte-modetnbert-base' — we extract embeddings from the selected terms. Then, we compare the candidate terms' embeddings with the most representative documents and rank them according to the obtained value of the cosine distance metric.

In the second step, we apply the Maximal Marginal Relevance (MMR) algorithm, which is a technique for query-based abstracting that maximizes the similarity of the query response fragments and minimizes the similarity to the fragments already selected in the response. Similar to TF-IDF, in the context of the current system, a term rather than a query is considered. That is, we maximize the internal diversity of the most representative term topics. MMR is computed using the following formula:

$$MMR = \underset{t_i\in\mathbb{R}\setminus\mathbb{S}}{\operatorname{argmax}}[\lambda(sim_1(t_i,\mathbb{C})) - (1-\lambda)\max_{t_j\in\mathbb{S}}(sim_2(t_i,t_j))], \tag{13}$$

where $\mathbb{T}$ is the topic for which MMR is calculated, $\mathbb{R}$ is the set of all terms t_i is the term under consideration, and $\mathbb{S}$ is the set of already selected terms. It turns out that we search for a new term from $mathbbR \setminus \mathbb{S}$ so that it is maximally similar to the topic, but minimally similar to the terms already present in the bag-of-words. In Equation 13, the coefficient λ is a hyperparameter

and balances the similarity of terms to the topic with the dissimilarity of terms to each other. The smaller the λ is, the less similar the terms are to each other, and the larger it is, the more similar the terms are to the topic.

Thus, we have a maximally representative set of terms, against which we can generate topic names. The system provides the use of text2text models directly for topic generation. Specifically, in the current implementation we used the GPT-4o model, for which we defined a corresponding prompt with instructions. The model was used with the help of OpenAI API. And all generated labels were later validated manually.

Finally, the final step is the visualization of the semantic map. The visualization was done in two formats: static — for work and presentation, and interactive — for system functioning. Visualization was performed using the Python library DataMapPlot, as well as custom add-ons in HTML, CSS and JavaScript. Also at the system level, functionality was developed for text search, building a word cloud, filtering by source, assets, publication date, and other quantitative attributes.

3.1 Invented Architecture

3.1.1 Overview

In this research, a new generation predictive system architecture has been developed. The main focus of this paper is the creation of a thematic modeling system and analytical interface for the financial subject area. A logical extension would be to build an interpretable and efficient predictive model based on the thematic tone of financial publications (news articles, press releases, posts, transcripts, etc.). Thus, the present work not only demonstrates the functioning principles of the basic module, but also formulates a vision of how it can be extended with asset price prediction capabilities.

The proposed architecture relies on proven robust mixed prediction approaches — a combination of convolutional and recurrent neural networks (CNN-LSTM) [Hochreiter, Schmidhuber, 1997; LeCun (et al.), 1998; Lu (et al.), 2020]. This solution allows simultaneous consideration of global long-term trends (such as the overall market trend) and local short-term patterns (such as head-and-shoulders, double bottoms, and other behavioral economics-driven structures).

It is important to keep in mind the variety of input data. Exchange metrics (Open, High, Low, Close, Volume, etc.) and derived technical indicators (RSI, MACD, etc.) are usually used for forecasting. Our study adds textual data as an Over The Counter (OTC) source. In the future, graph representations of links, images, audio and video recordings can be attracted as additional modalities. Thus, at the current stage we work with three modalities, each of which carries an independent semantic load and is able to explain price dynamics independently, but their interaction can both amplify the useful signal and introduce noise.

In this study, the architecture was based on CNN-LSTM using Slow Fusion. The main engineering challenge was to harmonize the spatio-temporal shapes of features of different modalities. To address this challenge, the architecture introduced a Feature Caching Mechanism block that synchronizes the tone vectors of the textual modality with the stock time series.

Additionally, the textual branch is preprocessed: based on the developed thematic modeling system, the thematic tones of publications are computed. This allows obtaining specialized scores for each topic, which gives an advantage over the use of a single common tone.

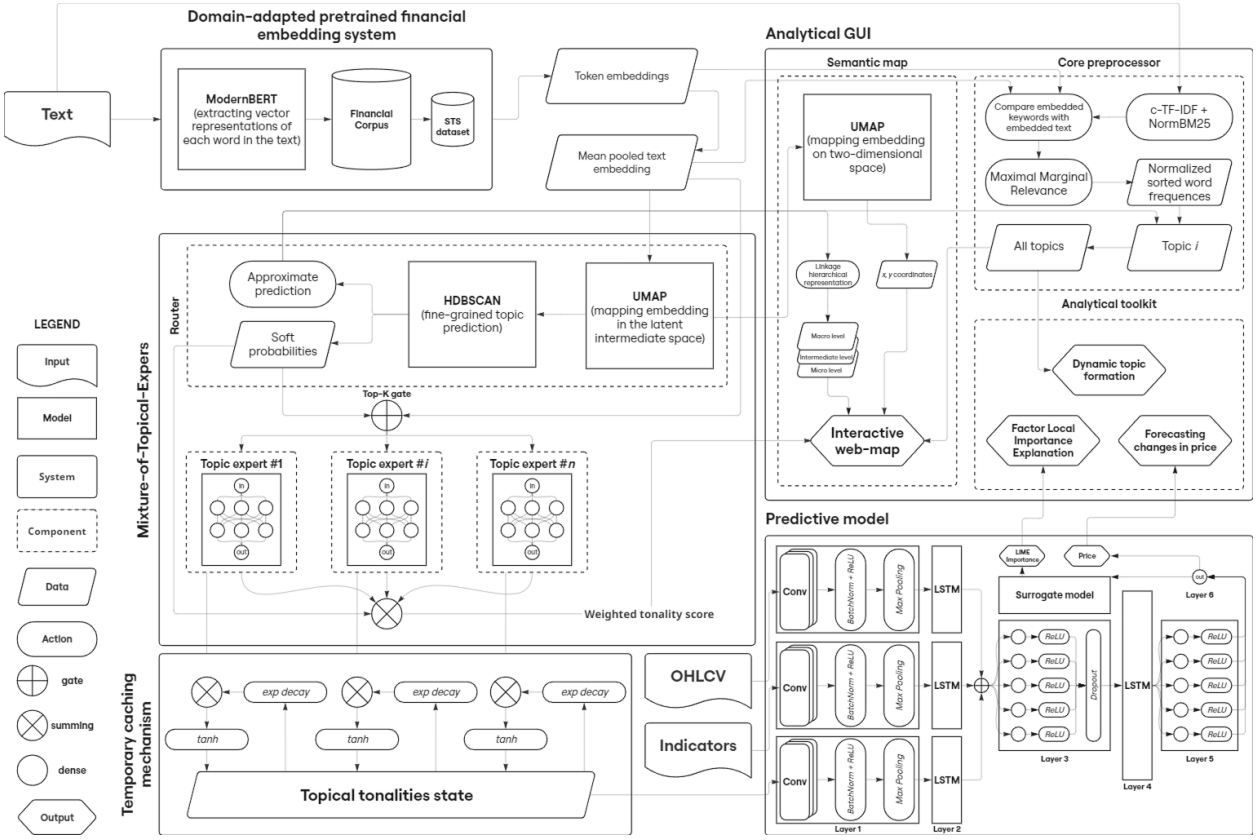

Figure 7: Overview of Financial Aspect-Based hybrid Semantic System (FinABYSS) architecture.

The designed system consists of five main modules (Figure 7):

- A financial language model (domain-adapted, fine-tuned for the STS task) that produces vector representations of the whole text and for each token.
- A Mixture-of-Topical-Experts that shapes the tone of the publication for each topic.
- A Temporal Caching Mechanism that aligns topic vectors to timestamps of stock data.
- A kernel predictive model with CNN-LSTM and Slow Fusion that processes all modalities and predicts asset price.
- An analytical GUI that aggregates intermediate results for further financial analysis.

Thus, the developed architecture is a powerful and adaptable tool for predicting the value of financial assets based on a comprehensive evaluation of the thematic sentiment of publications.

3.1.2 Embedding System

Proceeding to the consideration of individual blocks of the developed architecture, we should start with the stage of data preprocessing. The most resource-intensive and complex of them is the preparation of OTC text sources.

Immediately after entering the system, the text undergoes primary processing with the help of a domain-adapted financial embedding model configured for the task of semantic text comparison (STS). This model extracts vector representations for each token, thus preserving subtle contextual relationships within sentences. As new versions of embedding models become available, it is important to regularly update the underlying model and re-adapt it to the specifics of the financial domain.

The simplest path of domain adaptation involves an additional training stage on a Masked Language Modeling (MLM) task. The corpus of financial publications generated in this study or its extended version, as well as other relevant text corpora, can be used as a corpus for this purpose. For fine-tuning for the STS task, small labeled datasets are sometimes used, but to save resources it is possible to do without them, using contrastive learning methods on random subsamples of the main corpus [Gao, Yao, D. Chen, 2021].

Once the token embeddings are extracted, they are aggregated: usually averaging is applied, which gives a vector representation of the whole document. Then the document and its embeddings (both tokens and the averaged vector) are passed to the analysis module for visualization and interpretation of the results. In parallel, the document vector is sent to the "Mixture-of-Topical-Experts" block, where the topic tones of the publication are generated based on it.

As a concrete implementation, this study uses the 'gte-modernbert-base' model built on the ModernBERT architecture. Although it did not undergo domain adaptation initially, it shows high performance: the clustering of embeddings yielded a DBCV index of 0.485, and visual and contextual analysis showed a clear separation of thematic groups.

Thus, the proposed preprocessing step provides:

- Preserving deep semantic relations at the token and whole document level.
- Flexibility and reproducibility of the model adaptation process to the financial corpus.
- Integration of the results into the neighboring blocks of the architecture — a Mixture-of-Topical-Experts and analytical GUI system.

3.1.3 Mixture-of-Topical-Experts

This block utilizes a state-of-the-art Mixture-of-Experts (MoE) architecture that has been successfully applied in large language models (e.g., Mixtral 8×7B, DeepSeek R1) and in Google research (Switch Transformer) [Fedus, Zoph, Shazeer, 2022]. The main components, a learning router and multiple experts, allow only a small fraction of parameters to be dynamically activated at inference, ensuring high throughput and efficiency [Shazeer (et al.), 2017]. Especially for the analysis of financial texts, the experts are thematically specialized, which increases interpretability and accuracy in assessing the tone of publications.

MoE architecture is based on the principle of conditional computation: only a part of subnets ("experts") is active for each input example, which reduces computational costs at a huge total number of parameters [Shazeer (et al.), 2017].

- The router (gating network) receives the embedding of a document as input and computes a distribution of "weights" for all experts, then selects either the top-K experts or those whose weight exceeds a threshold [Fedus, Zoph, Shazeer, 2022].
- Experts are shallow feed-forward networks, each specializing in a different topic (in the context of the paper, one of the financial topics) [Shazeer (et al.), 2017].

When processing each embedding, only a small fraction of experts are activated — usually by Top K (selecting K of the most relevant experts) or by a threshold value (weighting the expert above a given threshold) — making MoE extremely computationally parsimonious with a colossal total number of parameters [Fedus, Zoph, Shazeer, 2022]. The values of K and θ act as hyperparameters and are customized on validation data.

For analyzing financial texts, the MoE model is a logical choice, since publications often contain a mixture of related topics (macroeconomics, corporate reports, geopolitics, etc.), and it is necessary to evaluate the tone from different angles. Experts make it possible to form specialized assessments for each topic simultaneously, preserving the "purity" of low-level processing and ensuring the interpretability of the results [Jacobs (et al.), 1991]. Nevertheless, it is also worth noting the high complexity of selecting K or θ hyperparameters [Ibid.].

Instead of training experts on subjectively labeled data, their parameters are optimized by back propagation of the error coming from the asset price prediction unit. This ensures that each expert is trained directly on the impact of publications on asset prices rather than on external annotations [Shazeer (et al.), 2017]. Once topic tones are obtained, they are aggregated by weighted summation considering topic probabilities, normalized, and sent to the analytics system for visualization, and to the feature caching engine for synchronization with exchange data.

Thus, the MoE-based Mixture-of-Topical-Experts block provides a combination of scalability, efficiency, and interpretability in analyzing financial texts. The dynamic activation of a small number of experts allows processing huge models without a proportional increase in computation, and training the experts through the signal from the price prediction model makes the estimation of tones objective and close to the economic reality. Thus, the proposed architecture opens up new opportunities for accurate prediction and in-depth analysis of the impact of textual publications on the value of financial assets.

3.1.4 Feature Caching Mechanism

The Feature Caching Mechanism is the central component of the architecture that ensures synchronization between exchange data (regularly arriving time series) and over-the-counter textual "ticks" scattered in time. When a new assessment of the tone of a publication on topic i arrives

from the Mixture-of-Topical-Experts block, it is instantly added to the matrix of accumulative exponentially smoothed values by Equation 14:

$$x_{i,t} = x_{i,t-1} + x_{i,0} \cdot e^{-\lambda \cdot t}, \tag{14}$$

where $x_{i,0}$ is the initial tone on topic i, $x_{i,t-1}$ is the previous smoothed value, and λ is the hyperparameter of the rate of exponential "fading" of the signal. This approach reflects the lagged and time-smoothed response of market participants to publications.

This approach takes into account that market participants' reaction to information events is delayed and distributed in time: the first "wave" of perception fades quickly, followed by smoother and longer effects, which in finance is traditionally modeled by exponential decline.

Since the original tones are normalized in the range $[-1,1]$, the cumulative summation by the Equation 14 can go beyond these bounds. To keep the scaling within acceptable intervals and to make the accumulation smooth, we propose to re-normalize the results using a hyperbolic tangent (Equation 15):

$$tone_{i,t} = \tanh(tone_{i,t-1} + x_{i,t}) = \frac{\tanh(tone_{i,t-1}) + \tanh(x_{i,t})}{1 + \tanh(tone_{i,t-1}) \cdot \tanh(x_{i,t})}. \tag{15}$$

which is equivalent to Equation 16:

$$tone_{i,t} = \frac{(1 - e^{-2 \cdot tone_{i,t}}) + (1 - e^{-2x_{i,t}} - (1 - e^{-2 \cdot tone_{i,t}}) \cdot (1 - e^{-2x_{i,t}})}{(1 + e^{-2 \cdot tone_{i,t}}) + (1 + e^{-2x_{i,t}} - (1 + e^{-2 \cdot tone_{i,t}}) \cdot (1 + e^{-2x_{i,t}})}, \tag{16}$$

where $tone_{i,t}$ is the current normalized exponentially smoothed tone on topic i at time t.

As a result of continuous operation of the Feature Caching Mechanism, a matrix of accumulative topic tonalities is formed and updated at each arrival of a textual "tick". While stock exchange data (Open, High, Low, Close, Volume and indicators) are absent, this time series accumulates "market sentiment" signals. As soon as a new time series of stock exchange measurements comes into the system (taking into account the network delay δ), the accumulated and normalized tones are concatenated with price and indicator data and passed to the Predictive Model for joint processing.

The dynamic process of the mechanism is illustrated in the Image 8. At the input moment, the cache is initialized with zero values, then it is updated with the first formula at each arrival of textual evaluation, after which it is normalized with a hyperbolic tangent. When a trigger is reached — moment t or stock data arrives at moment $t + \delta$ — the merged time series is passed on.

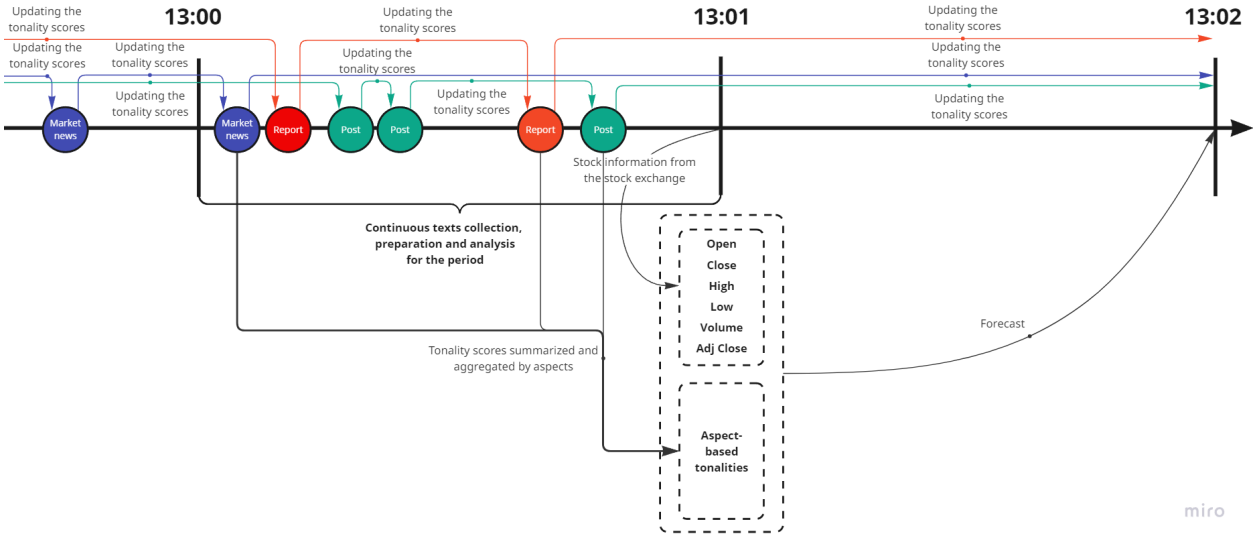

Figure 8: Dynamic representation of OTC data synchronization using Feature Caching Mechanism.

Thus, the Feature Caching Mechanism performs the following functions: converts irregular textual events into a continuous multidimensional time series, applies tone smoothing and scaling adequate to financial realities, and guarantees synchronization with stock exchange data. This ensures consistency and stability of input features in the Predictive Model and contributes to the accuracy of financial asset value forecasting.

3.1.5 Predictive Model

In the proposed hybrid architecture, we aim to combine the advantages of convolutional layers for local feature extraction and recurrent modules for modeling long-range dependencies, while providing adaptive fusion of quantitative and qualitative signals. The input is a multivariate time series of length T, in which each time step t corresponds to a vector $x_t = [x_t^{\text{price}}, x_t^{\text{ind}}, x_t^{\text{sent}}]$, where $x_t^{\text{price}} \in \mathbb{R}^{C_p}$ contains OHLCV, $x_t^{\text{ind}} \in \mathbb{R}^{C_i}$, and $x_t^{\text{sent}} \in \mathbb{R}^{C_s}$ is the set of sliding and smoothed values of news tones.

The data is first split into three branches: Price, Indicator, and Sentiment. Each branch undergoes two basic levels of processing. At the first level, in each branch, the sequence $\{x_t^{(\cdot)}\}_{t=1}^{T}$ is passed through a 1D convolution with small kernels and pooling operations (Equation 17)

$$y_t^{(\ell+1)} = \text{ReLU}\left(W^{(\ell)} * y_t^{(\ell)} + b^{(\ell)}\right), y_t^{(\ell+\frac{1}{2})} = \max\left(y^{(\ell+1)}t, y_{t+1}^{(\ell+1)}\right). \tag{17}$$

where the symbol $*$ denotes time convolution, the activation function ReLU is defined as follows (Equation 18):

$$\text{ReLU}(x) = \max(0, x). \tag{18}$$

The max pooling operation then halves the time dimensionality. This sequence allows the

initial layers to identify local patterns within each flow, be it short price swings, indicator pulses, or sentiment fluctuations.

Next, the output of the pooling in each branch is trained by a recurrent LSTM, which models the accumulation and forgetting of information taking into account long time dependencies. Let h_t and c_t be the hidden state and memory cell of the LSTM; their evolution is given by the standard input-, forget-, and output-gate equations. This ensures that each branch learns independently to determine which local features are significant for prediction at more distant time horizons.

After each branch has produced its hidden state h_t^{price}, h_t^{ind}, h_t^{sent} at each time step, an adaptive merging step takes place. For this purpose, the vector concatenation u_t is passed through the gate layer (Equation 19)

$$g_t = \sigma\big(W_g\, u_t + b_g\big). \tag{19}$$

where σ is a sigmoid and $g_t \in (0,1)^{\dim u}$ specifies the element-by-element coefficients controlling the relative contribution of each flux. The combined representation itself is computed as

$$m_t = g_t \odot u_t + (1 - g_t) \odot \mu(u_t). \tag{20}$$

where $\mu(u_t)$ can be an averaging or other aggregator of the components of the vector u_t. This mechanism allows the model to dynamically switch between emphasizing price patterns, indicator signals, or sentiment depending on the context of a particular time period.

The merged sequence $\{m_t\}_{t=1}^{T/2}$ is then fed back into the recurrent module merge-LSTM, which, like the previous ones, accumulates information about the joint development of the three modalities at a higher level of abstraction. Its output is the last hidden state $h_{T/2}^{\text{merge}}$, which contains a concise summary of the whole story. To add nonlinearity and an extra dimension of expressiveness, this state can be passed through a full-link layer with the ReLU function, and then through a Dropout layer to regularize and mitigate overtraining.

Finally, the final layer without activation translates the resulting vector into a univariate prediction $\hat{y}$, which is interpreted as a new price (in regression mode).

Also, it is worth noting that, for purposes of local interpretability of the tone estimates, an important role in the architecture is played by the presence of a separate surrogate model that, based on a sampling of a number of tones obtained from the FCM, with additional undulations, estimates the contribution of each topical tone to the final asset value prediction.

Thus, the proposed hybrid architecture combines local pattern detection (1D convolutions), the ability of LSTM to model long-term dependencies, and an adaptive fusion mechanism that allows dynamically adjusting the contribution of price, indicator, and sentiment signals. This provides high forecast accuracy with moderate computational complexity and transparency of the model with respect to the attributes used, while remaining lighter and more transparent than its transform-based counterparts.

3.1.6 Analytical Graphical User Interface

To conclude the description of the FinABYSS architecture, it is necessary to dwell on the key user component — the Analytical Graphical User Interface (GUI). This module accumulates the results of all previous stages: from pre-processing of OTC texts and their thematic interpretation in the Mixture-of-Topical-Experts block to concatenation with price and indicator series in the predictive model. The GUI inputs intermediate and final data, and its task is to bring them into a visual, interactive and analytically relevant form. The GUI consists of three interrelated components that implement data post-processing, provide deep analysis tools, and visualize semantic relationships between documents.

The first component, the Core Postprocessor, is responsible for transforming the "raw" outputs of the previous blocks into ordered and compact representations. In the initialization phase, it computes the relative frequencies of terms using a BM25-modification of TF-IDF and additional normalization, which removes noise and frequently occurring but semantically insignificant tokens (e.g., names of aggregators like Zacks). Next, after obtaining embeddings of tokens and the whole document from the domain-adapted language model, n-grams are ranked by frequency, taking into account the cosine similarity between the token vectors and the document vector. To ensure a balance between relevance and subsystem diversification, the Maximal Marginal Relevance (MMR) algorithm is applied: when partitioning into bigrams, MMR eliminates duplicate or redundantly close in meaning fragments, leaving the most informative ones. For example, from raw bigrams "ai" and "ai stocks", depending on the relative frequency and semantic proximity, either "ai" and "stocks", or only "ai stocks", or only "ai" can be retained, which increases the homogeneity of the final dictionary and reduces redundancy.

After MMR is applied, the dictionary is sorted by the resulting frequencies, and the postprocessor receives from the Mixture-of-Topical-Experts block the label of the cluster to which the current document belongs. Based on this cluster membership, incremental updating of c-TF-IDF statistics is performed: the accumulated word frequencies within each cluster are adjusted to reflect new publication, which provides adaptability and accounts for the evolution of thematic trends over time and is used for dynamic thematic modeling. Two data streams are fed in parallel from the postprocessor: the enriched frequency features and other extracted text metadata are sent to the Analytic Toolkit, and the cleaned and clusterred text with embeddings is sent to the Semantic Map.

The second component — Analytical Toolkit — is a set of visual and interactive elements that allow the user to explore in-depth the dynamics of thematic trends and their impact on asset prices. Through the dashboards it is possible to receive:

- historical tone charts for each topic, combined with price series and technical indicators;
- dynamic correlation matrices between news topics sentiments and price movements;

- summary reports on the impact of macro, meso and micro topics on volatility and trend movements;
- aggregated market sentiment indicators (weighted total tone across all topics) with trend lines.

The toolkit is designed with future expansion in mind: when adding new modalities (images, graph data, audio or video streams), modality-specific widgets can be easily integrated — for example, interactive clouds of key entities, graphs of relationships, or heat maps of sentiment in voice and video formats.

The third component, Semantic Map, implements two-dimensional visualization of document embeddings. Based on a pre-trained dimensionality reduction model, each document representation from the latent space where clustering was performed is mapped in (x, y) coordinates. In parallel, an approximate topic predictor from the Mixture-of-Topical-Experts block is used: for each document, a hierarchically organized triple of topic labels is defined — on macro-, meso- and micro-versions. The color, size, and shape of the markers on the map encode cluster membership, sentiment strength, and text volume. This allows to analyze not only the spatial neighborhood of publications, but also their topical multilevel structure, quickly identify peripheral and central documents, and track the evolution of topical communities.

Together, the GUI creates a single interactive panorama in which the results of the deepest computational building blocks — from the Feature Caching Mechanism and Mixture-of-Topical-Experts to the hybrid CNN-LSTM predictive model — are transformed into an intuitive and analytically rich tool. The user is able not only to view price forecasts, but also to explore in detail the cause-and-effect relationships between media signals and market movements.

The FinABYSS analytical GUI provides end-to-end visualization and interpretation of system output by integrating frequency, topic, and embedding attributes into three interconnected subsystems. The kernel postprocessor transforms raw textual and numerical data into compact and representative attributes, the analytical toolkit offers flexible tools for in-depth exploration of the impact of topics on asset value, and the semantic map provides a visualization of multidimensional embeddings based on hierarchical topics. With this combination of technological power and interoperability, the GUI is the final but no less important link in the forecasting pipeline, providing transparency, adaptability and visibility for financial experts.

3.2 Finished System Components

3.2.1 Embedding System and Router

The main result of this study was the creation of two key components: a flexible embedding subsystem and a trainable router for the MoTE unit. These developments not only provide a solid foundation for the further development of FinABYSS, but also open up a wide scope for use in related application projects.

As part of the study, we conducted a series of experiments to select optimal combinations of embedding models, dimension reduction methods, and clustering algorithms, using different metrics to adjust hyperparameters. At the first stage, the basic chain consisted of ModernBERT, UMAP (cosine metric), HDBSCAN (Euclidean metric) was optimized by the silhouette coefficient. Despite the satisfactory results, we noted a strong dependence on the cluster structure and the sensitivity of the silhouette to the shape of the clusters, which appeared to be arbitrary shapes in our data.

The second stage repeated the same bundle, but the DBCV index was chosen as the target metric. Due to the DBCV property of taking into account density features and heterogeneity of the point distribution, hyperparameters selected for DBCV provided more semantically consistent and dense clusters.

Finally, we tested base generally pre-trained [Warner (et al.), 2024] and the fine-tuned STS ("gte-modernbert-base") versions of ModernBERT [X. Zhang (et al.), 2024] in combination with UMAP, with cosine and L_2-Euclidean metric, and the same HDBSCAN, also configured for euclidean and L_2-Euclidean distance (Table 3). However, this combination was inferior to the previous one: the DBCV index was only 0.4032 against 0.4763 for the cosine + euclidean pair (Table 4).

Table 3: Description of the final configurations of experiments conducted on a subsample of 10,000 embeddings.

Model	**Configuration**		
	(I)	(II)	(III)
ModernBERT	Fine-tuned on STS	General pre-trained	Fine-tuned on STS
UMAP	L_2-Euclidean	Cosine	Cosine
HDBSCAN	L_2-Euclidean	Euclidean	Euclidean

Additional analysis of the noise points and cluster structure confirmed this advantage (Table 4). When using the L_2-Euclidean metric, the proportion of noise points reached 42.16%, and the number of clusters was 97 (with a maximum size of 3,220 points and a minimum size of 260). For the cosine + euclidean pair, the noise was only 36.18%, and the number of clusters increased to 162; at the same time, the maximum cluster increased to 3,742 points, and the minimum decreased to 109. Thus, the "cosine + euclidean" approach demonstrated a better ratio of cluster density and separability.

Table 4: Summary table of the results of hyperparameter optimization performed for three specified configurations on a subsample of 10,000 embeddings.

Best trial		**Configuration**		
		(I)	(II)	(III)
Hyperparameters	n_components	46	67	None
	n_neighbors	61	49	None
	min_dist	0,0435	0,0392	None
	spread	4,9	9,6	None
	negative_sample_rate	12	9	None
	min_cluster_size	260	102	None
	min_samples	191	108	None
Statistics	Max cluster size	3220	3742	None
	Min cluster size	260	109	None
	Total clusters	97	162	None
	Noise %	42,16%	**36,18%**	None
	DBCV Index	0,4032	**0,4766**	None

The experimental results clearly indicate the superiority of the ModernBERT + UMAP scheme with cosine metric and HDBSCAN with Euclidean distance in the tasks of topical clustering of financial texts. Combined with DBCV index optimization, this approach forms the most semantically coherent groups, minimizes the proportion of noise instances, and provides a more stable basis for subsequent training of the MoTE router.

Nevertheless, unfortunately, obvious problematic areas of the current implementation have also been identified. The resulting clusters and their structure cannot be incrementally updated with streaming new publications, which happens for several reasons. Firstly, the trained model is a GPU implementation from the cuML library, in which critical errors were discovered in the code version 25.02 [Raschka, Patterson, Nolet, 2020] only after conducting research. Secondly, the chosen UMAP algorithm does not work well enough in incremental mode by its nature, which is why there are other implementations, such as AlignedUMAP [McInnes, Healy, Saul, (et al.), 2018] or ParametricUMAP based on a neural network as a basic model [Sainburg, McInnes, Gentner, 2020]. Unfortunately, both of these implementations are available exclusively for training on a central rather than a GPU. And there are not enough computing resources for learning on a central processor in the context of the current study.

Thus, the bottleneck of this study and the proposed solution is the lack of computing power for training models, which will be improved in further research.

3.2.2 Semantic Map

Finally, based on the developed embedding subsystem and the MoTE router, a separate UMAP model was trained, designed to translate vectors from the intermediate latent clustering space into a two-dimensional representation convenient for visual analysis. It is this projection that underlies the Semantic Map of Financial Publications, one of the key components of the FinABYSS interface, which provides a deep and intuitive study of the thematic structure of news streams.

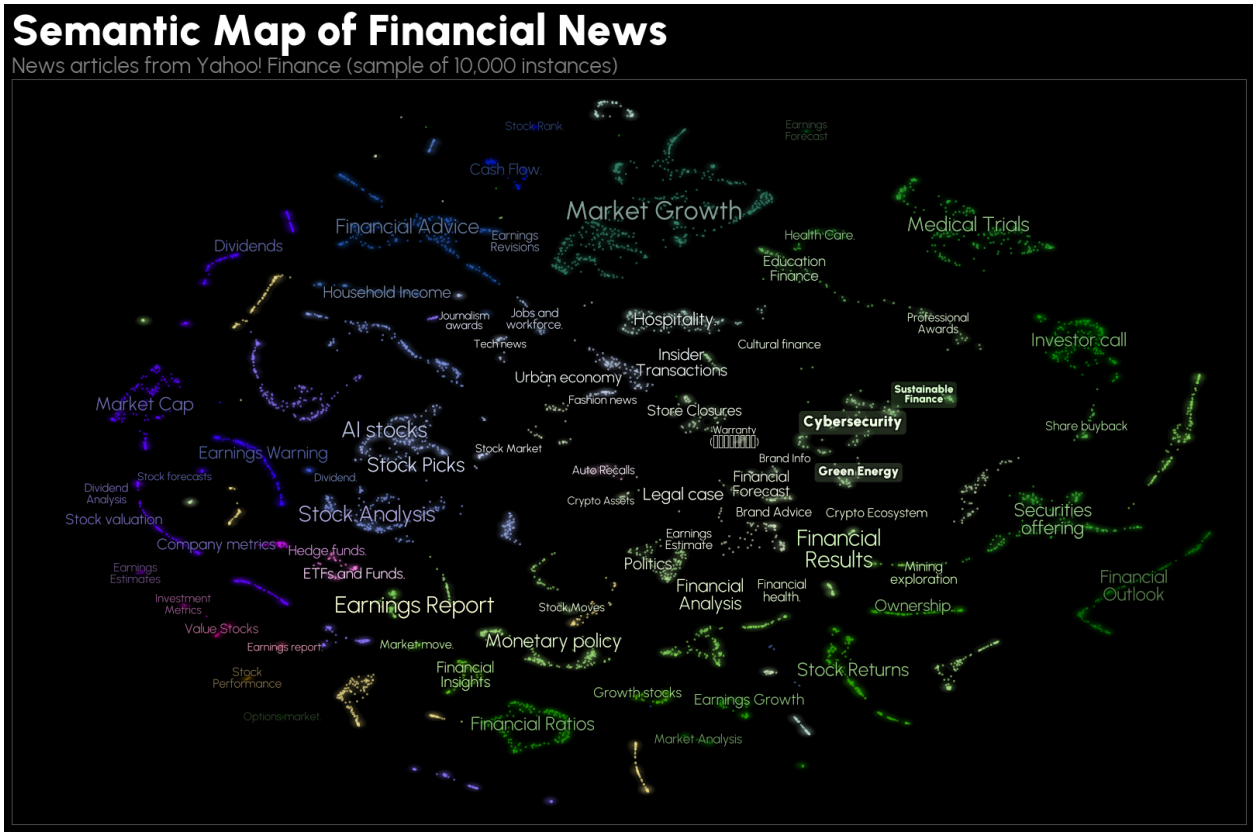

Figure 9: Semantic map (early demo version) of a sample of 10,000 financial publications from September 17, 2023 to March 18, 2025, clustered by financial topics.

First, a static version of the map was implemented, which displays 100,000 financial items from September 2023 to March 2025, divided into clusters by subject (Figure 9). Each cluster is marked with an automatically generated tag word without any manual annotation. Despite the automatic nature of the assignment, the labels turned out to be representative: dense groups of articles on healthcare, "Sustainable Finance," "Cybersecurity," and "Green Energy" are located side by side, reflecting their semantic proximity. The same thing happens with the "Politics" and "Monetary Policy" clusters.

However, the static map only demonstrates the potential of UMAP projection. The FinABYSS interactive Semantic map goes far beyond the simple visualization of points:

- Hovering over any point reveals the metadata of the article: title, date and time of publica-

tion, hierarchical topic (macro/meso/microtheme), author and source. At the same time, a preview of the full text and a direct link to the original publication are available.

- The keyword search engine allows you to quickly filter articles containing special terms. The search can be combined with filters by date range, publication volume, and other numerical attributes (for example, text volume or number of views), which makes it easier to spot historical events and triggers on the graph.
- The Sources and Topics section makes it possible to include and exclude news sources or target clusters, helping to focus on relevant publications in complex analytical scenarios.
- A word cloud function is also available for articles, which is formed based on the most frequent words found in a selected group of texts. The word cloud instantly displays the dominant terms and discussion patterns, which complements the quantitative outline of the graph with qualitative characteristics.

Thus, the FinABYSS semantic map is a full-fledged analytical system that combines the power of the selected UMAP and HDBSCAN models, and with further development, the hybrid CNN-LSTM architecture and the MoE approach to sentiment analysis. It provides the researcher with the opportunity not only to visually distinguish named clusters and their mutual arrangement, but also to delve deeply into the content of each publication, combining automatic and manual analysis methods.

The developed interactive Semantic map is a natural continuation of the previous FinABYSS modules. All the pipeline links are connected in a single chain. The tool offers the user a transparent, scalable and flexibly customizable interface for semantic research of financial news, where each element — from cluster labels to a cloud of words — reflects the results of the computational logic of the system and supports expert solutions in the analysis of market processes.

3.2.3 Dynamic Topic Modeling

In addition to the Semantic Map, FinABYSS provides powerful tools for analyzing the linguistic features of formed thematic groups and dynamic temporal modeling of topics. All incoming texts go through the post-processing pipeline described in Section 2.4 in the context of the Analytical GUI (see Section 3.1.6), where each document is assigned a prevailing theme, and then the corresponding lexical features are extracted and aggregated.

Thus, the system visualizes the frequency distribution of the most relevant words within the selected topic (Figure 10). This linguistic panel serves two purposes:

- Firstly, after developing the system, it allows to validate the quality of the generated topics and assess how unique and semantically homogeneous they are, in manual mode.

- Secondly, this functionality can be very useful when a financial analyst is first introduced to the system. A financial analyst working with FinABYSS for the first time gets an instant understanding of the content of each topic without deep reading of texts.

It was the latter fact that led to the inclusion of this functionality in the set of basic analytical GUI toolkit.

Figure 10: Topics words frequencies distribution

To illustrate this functionality, a subsample of 100,000 publications was randomly generated and 12 topics were selected within them for display. Among the sample, all topics undoubtedly have practical significance for financial markets. Moreover, there are more general industry topics such as:

- Topic 2 related to the pharmaceutical industry;
- Topic 13 related to the artificial intelligence sector;
- Topic 28 related to cybersecurity;
- Topic 33 related to the mining industry;
- Topic 54, related to the aviation industry.

On the other hand, the sample also includes a common topic that is on the agenda in finance — Topic 56, which clearly relates to ESG. Finally, there are highly specialized financial topics that were also identified automatically:

- Topic 4 related to lawsuits and proceedings, which is potentially extremely important and influential in the context of asset pricing;
- Topic 6 related to external economic factors and the economy in general;
- Topic 11, directly related to the asset market, namely cryptocurrencies;
- Topic 15 related to company ownership rights;
- Topic 21 related to the foreign exchange market;
- Topic 22 related to cash flows, including future, present and discounted cash flows.

So, we can observe the linguistic features of the topics, as well as very quickly and effectively study both the differences between them and specific topics in depth.

On the other hand, it would be extremely useful to understand how topics change over time, because the information media space is extremely unstable, and the agenda in modern society is changing extremely quickly. Thus, FinABYSS implements dynamic thematic modeling through an interactive graph of topical time series.

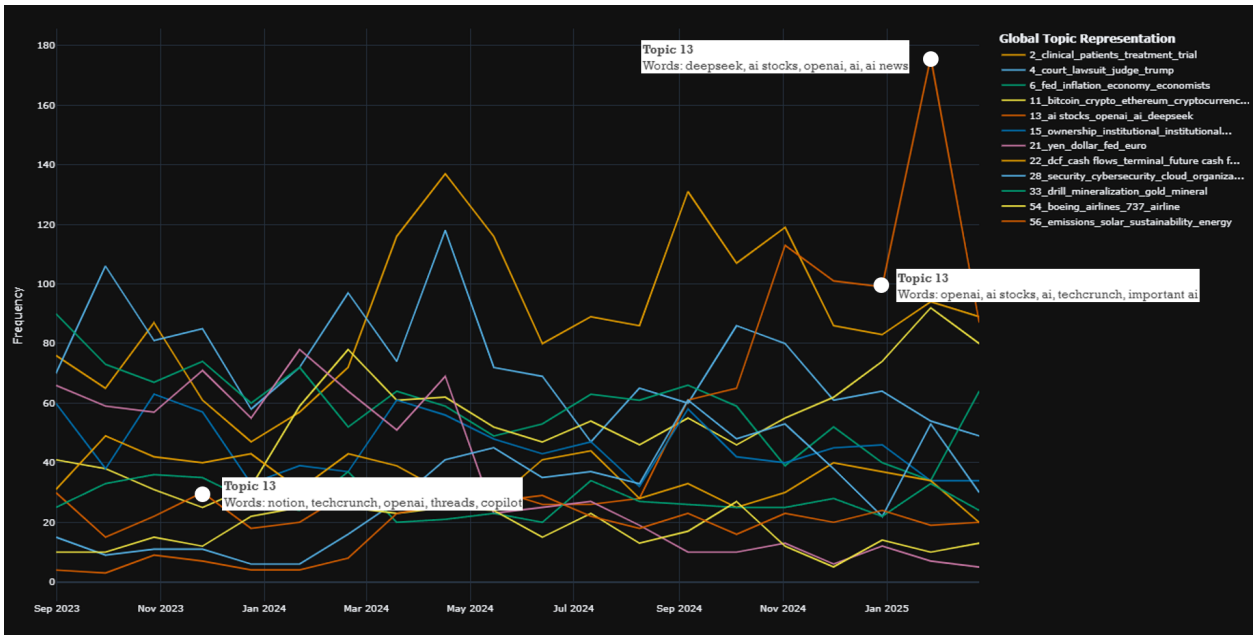

Figure 11: Dynamic topic modeling

The Figure 11 shows the same subsample of 12 topics. The abscissa axis reflects the publication interval (configurable: from daily to monthly or annual), the ordinate axis is the number of articles on each topic for the corresponding period. When hovering over the cursor, the five most representative words describing the topic in this time window are displayed. The user can change the number of words displayed.

The advantages of this approach are obvious:

- Early detection of trends. The analyst can observe the appearance of new lexical markers or the growth of publication activity in thematic groups, which serves as a signal of emerging events.
- Tracking cyclical phenomena. So, for Topic 13 (AI), the word TechCrunch appears on the chart in December 2023 and December 2024, the name of one of the largest and most prestigious conferences in the fields of IT and AI. Every year, the conference produces many of the largest startups, which later receive large amounts of funding. Thus, with this visualization, the analyst can keep up to date with cyclical events that are significant for finance with less effort.
- Reaction to force majeure events. In February 2025, there is a sharp peak on the same Topic 13 due to the announcement of the Chinese LLM DeepSeek R1. This incident is extremely important, as it subsequently caused the collapse of Nvidia shares, the largest company supplying computing equipment.

Thus, FinABYSS is not limited to the static construction of thematic clusters: interaction with linguistic metrics and temporal trends turns the system into a universal platform for financial analytics. The expert can switch from macro-trends to micro-lexical details in a few clicks, combine filters by date, source and subject, study the evolution of terminology and quickly respond to the appearance of new keywords or abnormal frequency changes. This makes FinABYSS not just a clustering tool, but a full-fledged ecosystem for semantic monitoring and forecasting the impact of media signals on the dynamics of financial assets.

3.2.4 Practical Importance

To summarize, it is worth emphasizing that the FinABYSS system is not just a set of models, but a full-fledged tool for rapid detection of critical signals in financial markets. Semantic map, deep topical clustering and news time series allow financial analysts to instantly react to unexpected events, reduce risks and profit.

Thus, looking solely at the functionality of the Semantic Map, one can find several critical events that were immediately reflected in the “Legal case” thematic cluster in a timely manner after they hit the web.

The first illustrative example (Figure 12) demonstrates how an article dated May 22, 2024 affected the stock price of the target company. This news highlighted the scandal of the collapse of the value of European stocks due to insufficient control over trading operations by one of the 4 largest banks in the world — Citi Group (C.NYSE) — whose shares are traded on the New York Stock Exchange. After this news, which also announced the imposition of a record fine on the bank by the British government, the company’s shares fell almost 4% over the next 28 hours. Over the next 23 days, as litigation, additional reports and delayed market reaction took place, the price fell nearly 10%.

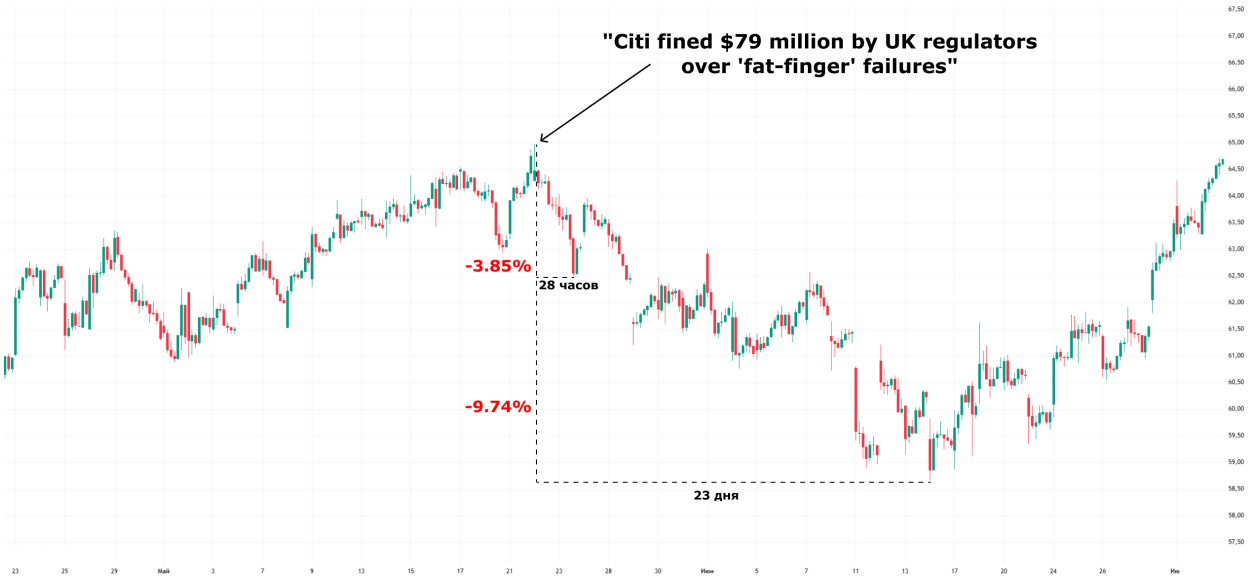

Figure 12: An example of the fall in the share price of one of the largest US banks, Citi Group (C.NYSE), due to allegations of insufficient control over trading operations, which caused a fall in European shares

The Reuters news article itself signaled a sell signal, while the Figure 12 clearly shows a tipping point with a sharp drop in price, a small pullback and a further change in the global trend for almost a month.

Without FinABYSS, such signals are often lost in the news flow, while the developed system simplifies the detection of trigger events by implementing the ability to track clusters significant for a particular portfolio.

The second case study relates to a news item dated August 30, 2024 about a possible license revocation and AU$67 million fine against Australia's largest gambling company, The Star Entertainment Group LTD (Figure 13). News reported allegations of money laundering. The shares were frozen for a month, and when trading resumed, their value collapsed by 55%.

Figure 13: An example of Australian company The Star Entertainment Group LTD (SGR.ASX) shares collapsing and being temporarily halted from trading due to money laundering litigation.

It is important to note that trading was halted 24 hours after the news was published, which is a sufficient interval for a sell signal to be detected. However, for the average investor not involved in intraday trading, this signal would very likely have been undetectable, leading to the dire situation of an investor unable to sell a depreciating asset and forced to hold an obvious loss indefinitely.

In both cases, with FinABYSS, a financial analyst or investor would be among the first to recognize these incidents and react immediately to market signals. The simplest way to learn about triggers in a timely manner is the asset-based tracking module, which allows you to set filters by specific ticker, source, and topic to receive only targeted signals.

Eventually, once the architecture proposed in Section 3.1 is fully implemented and trained, it will be possible to customize the filtering of news by its sentiment, as well as setting a custom threshold to detect meaningfully positive or negative news.

FinABYSS thus takes financial analytics to a whole new level: instead of haphazard monitoring of media and articles, it gets ready-to-action alerts, visualization, and predictive support. The semantic map is already becoming an integral part of the workflow, and its value will only grow with the expansion of modalities and the introduction of a predictive mechanism.

3.3 Semantic De-duplication Solution

3.3.1 Mathematical Formulaion

Within the framework of this study, a novel deduplication approach was developed based on the analysis of the semantic content of objects. Although in the present work the entity is a text, the method can be readily generalized to any objects that admit a vector representation in a

semantic space.

Each article is represented as a sequence of embeddings:

$$x_i \subset \mathbb{R}^{t \times d} \tag{21}$$

where t is the number of tokens and d is the dimensionality of the semantic vector space. For subsequent analysis, instead of the raw set of embeddings, their convex hull is used, denoted as $\mathrm{CH}(x_i)$ or, for brevity, CH_i. The uniqueness of a text is quantified by the volume of this convex hull, $\mathrm{vol}(\mathrm{CH}_i)$.

Accounting for the Intersections of Convex Hulls. Direct subtraction of the intersections between CH_i and the hulls of other texts may lead to multiple counting. To eliminate this issue, the inclusion–exclusion principle is applied.

Let the set of all articles except i be denoted by

$$\mathbb{I} = \{1, \ldots, N\} \setminus \{i\}. \tag{22}$$

The intersection of CH_i with the hulls of articles indexed by subsets $\mathbb{J} \subseteq \mathbb{I}$ is expressed as:

$$\mathrm{vol}\Big(\mathrm{CH}_i \cap \bigcap_{j \in \mathbb{J}} \mathrm{CH}_j\Big). \tag{23}$$

Then, the volume of the intersection of CH_i with the union of the hulls of the remaining articles is computed as:

$$\mathrm{vol}\left(\mathrm{CH}_i \cap \bigcup_{j \in \mathbb{I}} \mathrm{CH}_j\right) = \sum_{k=1}^{N-1} (-1)^{k-1} \sum_{\substack{\mathbb{J} \subseteq \mathbb{I} \\ |\mathbb{J}|=k}} \mathrm{vol}\left(\mathrm{CH}_i \cap \bigcap_{j \in \mathbb{J}} \mathrm{CH}_j\right). \tag{24}$$

The uniqueness of article i is defined as the fraction of its convex hull's volume that is not occupied by intersections with the hulls of other articles:

$$\mu_i = \frac{\mathrm{vol}\Big(\mathrm{CH}_i \setminus \bigcup_{j \in \mathbb{I}} \mathrm{CH}_j\Big)}{\mathrm{vol}\Big(\mathrm{CH}_i\Big)}. \tag{25}$$

By decomposing CH_i into the intersection region and its complement, we obtain:

$$\mu_i = 1 - \frac{\mathrm{vol}\left(\mathrm{CH}_i \cap \bigcup_{j \in \mathbb{I}} \mathrm{CH}_j\right)}{\mathrm{vol}\left(\mathrm{CH}_i\right)}. \tag{26}$$

Substituting the inclusion–exclusion formulation 24, the final expression becomes:

$$\mu_i = 1 - \frac{1}{\mathrm{vol}\big(\mathrm{CH}_i\big)} \sum_{k=1}^{N-1} (-1)^{k-1} \sum_{\substack{\mathbb{J} \subseteq \mathbb{I} \\ |\mathbb{J}|=k}} \mathrm{vol}\Big(\mathrm{CH}_i \cap \bigcap_{j \in \mathbb{J}} \mathrm{CH}_j\Big). \tag{27}$$

The value $\mu_i \in [0,1]$ characterizes the text's uniqueness: $\mu_i = 1$ indicates no intersections with other texts (complete uniqueness), while $\mu_i = 0$ implies that the semantic volume of the text is entirely occupied by intersections with the hulls of other texts.

3.3.2 Pros and Cons

The proposed method is founded on a theoretically sound representation of text: each article is treated as the convex hull of its token embeddings. This representation enables a precise definition of an object's semantic content, facilitates the application of the inclusion-exclusion principle (inherited from set theory) for accurate calculation of intersection volumes, and normalizes the result so that the final uniqueness measure lies within the interval $[0,1]$.

In addition to its theoretical rigor, the method offers several advantages:

- The use of embeddings for each token allows for capturing subtle distinctions in semantic content, while aggregation via the convex hull yields a generalized representation of the text. This approach enables the comparison of texts of varying lengths and topics within a unified vector space.
- Normalization of the metric to a $[0,1]$ interval simplifies interpretation.

Conversely, the method has several notable drawbacks:

- In high-dimensional spaces (e.g., 768 dimensions), the convex hull may become excessively "stretched," resulting in uninformative volume measurements, and its geometry may fail to accurately reflect the complex distribution of embeddings.
- Embeddings typically possess a complex, often non-linear structure. As the convex hull is the minimal convex set containing the data, it may enclose extreme points, leading to an overestimation of the occupied space and, consequently, to skewed evaluations.
- Since embeddings can include random noise or artifacts, the convex hull is sensitive to outliers. Minor inaccuracies in embeddings may disproportionately enlarge the convex hull's volume, thus distorting the uniqueness assessment.
- Constructing the convex hull and computing volumes in high-dimensional spaces is computationally intensive. Moreover, applying the inclusion–exclusion principle to accurately compute intersections between the hulls of texts further complicates calculations, particularly with a large number of documents.

These challenges can critically affect the practical application of the method; however, some can be mitigated through engineering solutions.

The sensitivity to noise (item 3) can be partially alleviated by using the [CLS] token as the centroid of the convex hull. Introducing a coefficient δ to normalize the "concavity" of the hull in the direction of the [CLS] embedding helps to diminish the impact of noisy components.

The computational burden (item 4) can be addressed through various strategies:

- Regulating the number of inclusion–exclusion pairs (the hyperparameter N in the summation) allows for an approximate evaluation of uniqueness while reducing computational demands.
- Employing dimensionality reduction algorithms, such as UMAP, t-SNE, or PCA, can project the original space onto a lower-dimensional one, substantially decreasing computational costs, though potentially at the expense of some accuracy.
- Approximating the volume using Monte Carlo methods offers an alternative that lessens computational load.

The method of representing semantic uniqueness of text via the convex hulls of embeddings boasts several theoretical advantages (robust normalization, applicability of the inclusion–exclusion principle, and consistent interpretability of the result). Nevertheless, its practical deployment necessitates addressing challenges related to high dimensionality, non-linear distributions of embeddings, and substantial computational costs. Future research may focus on developing more robust and computationally efficient methods for assessing text uniqueness while accommodating these limitations.

Conclusion

Disparity in Access to Financial Resources. During the study, it was found that there exists a significant barrier for individual researchers who lack the financial resources required for expensive data collection, infrastructure rental, and the time needed to develop a system entirely from scratch.

The financial community — which includes news outlets, data aggregators, professional traders, and investment funds — often does not facilitate the development of publicly available tools for extracting value from financial instruments. On the contrary, several market participants deliberately create additional obstacles to free data access, while failing to utilize existing resources efficiently. Examples include:

- **Infrastructure limitations.** Restrictions imposed by aggregators and news services (e.g., Yahoo! Finance) impede large-scale data collection.
- **Closed APIs and high tariffs.** Services such as Google Finance and Yahoo! Finance, along with platforms like Twitter and Seeking Alpha, offer limited functionality or charge high fees for access.
- **Restrictions on access to analytical tools.** Cases such as BloombergGPT illustrate the deliberate concealment of general-purpose tools.
- **Strict copyright policies.** Tighter copyright conditions result in restricted access to various datasets [Wu (et al.), 2023].

Thus, it can be concluded that the financial community contributes to a scarcity of open informational resources by artificially raising the barriers to access with the aim of reducing competition and limiting the number of independent market players.

This issue is not new — it has been repeatedly highlighted in several studies (including by the creators of FinBERT [Y. Yang, UY, A. Huang, 2020]); however, over the past five years the situation has remained virtually unchanged. A crisis also persists in the open-source segment of financial tools.

Despite the widespread restrictive practices, there are proactive participants in the financial sector who strive to distribute information more equitably. For instance, the financial data provider Alpha Vantage[24] offers a free and open API that grants access to a vast array of valuable data, including intraday OHLCV. Although Reddit[25] is less popular than platform X (ex-Twitter)[26] in the financial community, it also provides an open API and can serve as an alternative channel for publishing announcements, opinions, and insider information.

[24]URL: `https://www.alphavantage.co/`
[25]URL: `https://www.reddit.com/`
[26]URL: `https://x.com/`

In addition, aggregators such as FinURLs[27] and MarketWatch[28] represent important information sources. FinURLs compiles links to historical news from 24 sources over several years. Despite the lack of a dedicated API and certain interface inconveniences for data extraction, this resource remains valuable. At the same time, MarketWatch boasts a more advanced infrastructure by offering not only links to news articles but also quantitative data, as well as the ability to obtain information on specific markets, assets, or indices.

Individual yet significant sources, such as the websites of certain companies and government agencies, also deserve attention. For example, the SEC[29] provides free access to historical financial reports (e.g., 10-K and 10-Q) via an RSS feed, thereby promoting more equitable access to information. However, even these open datasets are frequently accompanied by technical challenges: precise timestamps are often missing or the website structure is disrupted, which complicates automated data extraction.

It should be noted that nearly all real-time data are available without significant restrictions, as most services promptly provide such information. Nevertheless, the collection of both historical and real-time data regularly encounters ethical and copyright issues, which remain an important aspect in the practical use of these resources.

In summary, despite various initiatives aimed at expanding access to financial data, the overall landscape is still characterized by artificially high barriers. These restrictions contribute to a shortage of open tools, which in turn reduces market competition and limits opportunities for independent researchers. Therefore, the development of methodologies aimed at the free and equitable dissemination of information remains an urgent task, requiring a comprehensive approach that takes technical, ethical, and legal aspects into account.

References

Akiba, *T.* Optuna: A Next-Generation Hyperparameter Optimization Framework / T. Akiba [et al.] // The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. — 2019. — P. 2623–2631.

Alvarado, *J. C. S.* Domain adaption of named entity recognition to support credit risk assessment / J. C. S. Alvarado, K. Verspoor, T. Baldwin // Proceedings of the australasian language technology association workshop 2015. — 2015. — P. 84–90.

Amid, *E.* TriMap: Large-scale Dimensionality Reduction Using Triplets / E. Amid, M. K. Warmuth // arXiv preprint arXiv:1910.00204. — 2019. — arXiv: 1910.00204.

Araci, *D.* FinBERT: Financial Sentiment Analysis with Pre-trained Language Models / D. Araci. — 2019. — Aug. — URL: http://arxiv.org/abs/1908.10063.

[27] URL: https://finurls.com/

[28] URL: https://www.marketwatch.com/

[29] URL: https://www.sec.gov/

Au, *W.* FinSBD-2021: The 3rd Shared Task on Structure Boundary Detection in Unstructured Text in the Financial Domain / W. Au, A. Ait-Azzi, J. Kang // Companion Proceedings of the Web Conference 2021. — Ljubljana, Slovenia : Association for Computing Machinery, 2021. — P. 276–279. — (WWW '21). — ISBN 9781450383134. — DOI: 10.1145/3442442.3451378. — URL: https://doi.org/10.1145/3442442.3451378.

Auger, *A.* A restart CMA evolution strategy with increasing population size / A. Auger, N. Hansen // 2005 IEEE congress on evolutionary computation. Vol. 2. — IEEE. 2005. — P. 1769–1776.

Bentivogli, *L.* The Fifth PASCAL Recognizing Textual Entailment Challenge. / L. Bentivogli [et al.] // TAC. — 2009. — Vol. 7, no. 8. — P. 1.

Bergstra, *J.* Algorithms for hyper-parameter optimization / J. Bergstra [et al.] // Advances in neural information processing systems. — 2011. — Vol. 24.

Bogdan, *R.* Qualitative research for education. Vol. 368 / R. Bogdan, S. K. Biklen. — Allyn & Bacon Boston, MA, 1997.

Campello, *R. J.* Density-based clustering based on hierarchical density estimates / R. J. Campello, D. Moulavi, J. Sander // Pacific-Asia conference on knowledge discovery and data mining. — Springer. 2013. — P. 160–172.

Cer, *D.* SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation / D. Cer [et al.] // Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). — Association for Computational Linguistics, 2017. — DOI: 10.18653/v1/s17-2001. — URL: http://dx.doi.org/10.18653/v1/S17-2001.

Chen, *Z.* Quora question pairs. / Z. Chen [et al.]. — 2017.

Dao, *T.* FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness / T. Dao [et al.]. — 2022. — arXiv: 2205.14135 [cs.LG]. — URL: https://arxiv.org/abs/2205.14135.

Daudert, *T.* A multi-source entity-level sentiment corpus for the financial domain: the FinLin corpus / T. Daudert // Language Resources and Evaluation. — 2022. — Vol. 56, no. 1. — P. 333–356.

Devlin, *J.* Bert: Pre-training of deep bidirectional transformers for language understanding / J. Devlin [et al.] // 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. — P. 4171–4186.

Dolan, *W. B.* Automatically Constructing a Corpus of Sentential Paraphrases / W. B. Dolan, C. Brockett // Proceedings of the Third International Workshop on Paraphrasing (IWP2005). — 2005. — URL: https://aclanthology.org/I05-5002/.

Du, *K.* Financial Sentiment Analysis: Techniques and Applications / K. Du [et al.] // ACM Computing Surveys. — 2024. — Oct. — Vol. 56, issue 9. — ISSN 15577341. — DOI: 10.1145/3649451.

Dudy, *S.* A Multi-Context Character Prediction Model for a Brain-Computer Interface / S. Dudy [et al.] // Proceedings of the Second Workshop on Subword/Character LEvel Models / ed. by M. Faruqui [et al.]. — New Orleans : Association for Computational Linguistics, 06/2018. — P. 72–77. — DOI: 10.18653/v1/W18-1210. — URL: https://aclanthology.org/W18-1210/.

Dutt, *A.* Shared manifold learning using a triplet network for multiple sensor translation and fusion with missing data / A. Dutt, A. Zare, P. Gader // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. — 2022. — Vol. 15. — P. 9439–9456.

Ester, *M.* A density-based algorithm for discovering clusters in large spatial databases with noise / M. Ester [et al.] // kdd. Vol. 96. — 1996. — P. 226–231.

Falkner, *S.* BOHB: Robust and efficient hyperparameter optimization at scale / S. Falkner, A. Klein, F. Hutter // International conference on machine learning. — PMLR. 2018. — P. 1437–1446.

Fama, *E. F.* Efficient Capital Markets: A Review of Theory and Empirical Work / E. F. Fama // The Journal of Finance. — 1970. — Vol. 25, no. 2. — P. 383–417. — ISSN 00221082, 15406261. — URL: http://www.jstor.org/stable/2325486 (visited on 04/30/2025).

Fedus, *W.* Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity / W. Fedus, B. Zoph, N. Shazeer // Journal of Machine Learning Research. — 2022. — Vol. 23, no. 120. — P. 1–39.

Feichtenhofer, *C.* Convolutional two-stream network fusion for video action recognition / C. Feichtenhofer, A. Pinz, A. Zisserman // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 1933–1941.

Gao, *T.* Simcse: Simple contrastive learning of sentence embeddings / T. Gao, X. Yao, D. Chen // arXiv preprint arXiv:2104.08821. — 2021.

Grootendorst, *M.* BERTopic: Neural topic modeling with a class-based TF-IDF procedure / M. Grootendorst // arXiv preprint arXiv:2203.05794. — 2022.

Gururangan, *S.* Don't stop pretraining: Adapt language models to domains and tasks / S. Gururangan [et al.] // arXiv preprint arXiv:2004.10964. — 2020.

Halder, *S.* FinBERT-LSTM: Deep Learning based stock price prediction using News Sentiment Analysis / S. Halder. — 2022. — Nov. — URL: http://arxiv.org/abs/2211.07392.

Hochreiter, *S.* Long Short-Term Memory / S. Hochreiter, J. Schmidhuber // Neural Computation. — 1997. — Nov. — Vol. 9, issue 8. — P. 1735–1780. — ISSN 08997667. — DOI: 10.1162/neco.1997.9.8.1735.

Howard, *J.* Universal language model fine-tuning for text classification / J. Howard, S. Ruder // arXiv preprint arXiv:1801.06146. — 2018.

Huang, *A.* FinBERT: A Large Language Model for Extracting Information from Financial Text* / A. Huang, H. Wang, Y. Yang // Contemporary Accounting Research. — 2023. — May. — Vol. 40, issue 2. — P. 806–841. — ISSN 19113846. — DOI: 10.1111/1911-3846.12832.

Huang, *H.* Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization / H. Huang [et al.] // Communications biology. — 2022. — Vol. 5, no. 1. — P. 719.

Jacobs, *R.* Adaptive Mixtures of Local Experts / R. Jacobs [et al.] // Neural Computation. — 1991. — Mar. — Vol. 3. — P. 79–87. — DOI: 10.1162/neco.1991.3.1.79.

Jain, *A. K.* Data clustering: 50 years beyond K-means / A. K. Jain // Pattern recognition letters. — 2010. — Vol. 31, no. 8. — P. 651–666.

Jiang, *T.* Financial sentiment analysis using FinBERT with application in predicting stock movement / T. Jiang, A. Zeng. — 2023. — June. — URL: http://arxiv.org/abs/2306.02136.

Joze, *H. R. V.* MMTM: Multimodal transfer module for CNN fusion / H. R. V. Joze [et al.] // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2020. — P. 13289–13299.

Karpathy, *A.* Large-scale Video Classification with Convolutional Neural Networks / A. Karpathy [et al.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 06/2014.

Kim, *J.* Forecasting the S&P 500 Index Using Mathematical-Based Sentiment Analysis and Deep Learning Models: A FinBERT Transformer Model and LSTM / J. Kim, H. S. Kim, S. Y. Choi // Axioms. — 2023. — Sept. — Vol. 12, issue 9. — ISSN 20751680. — DOI: 10.3390/axioms12090835.

LeCun, *Y.* Gradient-based learning applied to document recognition / Y. LeCun [et al.] // Proceedings of the IEEE. — 1998. — Vol. 86, no. 11. — P. 2278–2324.

Levesque, *H. J.* The Winograd schema challenge. / H. J. Levesque, E. Davis, L. Morgenstern // KR. — 2012. — Vol. 2012. — 13th.

Li, *L.* Hyperband: A novel bandit-based approach to hyperparameter optimization / L. Li [et al.] // Journal of Machine Learning Research. — 2018. — Vol. 18, no. 185. — P. 1–52.

Li, *Z.* Towards general text embeddings with multi-stage contrastive learning / Z. Li [et al.] // arXiv preprint arXiv:2308.03281. — 2023.

Liaw, *R.* Tune: A Research Platform for Distributed Model Selection and Training / R. Liaw [et al.] // arXiv preprint arXiv:1807.05118. — 2018.

Liu, *Z.* FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining : tech. rep. / Z. Liu [et al.]. — 2020. — URL: http://commoncrawl.org/.

Lu, *W.* A CNN-LSTM-based model to forecast stock prices / W. Lu [et al.] // Complexity. — 2020. — Vol. 2020, no. 1. — P. 6622927.

Macedo, *M.* WWW'18 Open Challenge: Financial Opinion Mining and Question Answering / M. Macedo [et al.] // Companion Proceedings of The Web Conference 2018. — Lyon, France : International World Wide Web Conferences Steering Committee, 2018. — P. 1941–1942. — (WWW '18). — ISBN 9781450356404. — DOI: 10.1145/3184558.3192301. — URL: https://doi.org/10.1145/3184558.3192301.

Malo, *P.* Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts / P. Malo [et al.] // Journal of the Association for Information Science and Technology. — 2014. — Vol. 65.

Malzer, *C.* A Hybrid Approach To Hierarchical Density-based Cluster Selection / C. Malzer, M. Baum // 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). — IEEE, 09/2020. — P. 223–228. — DOI: 10.1109/mfi49285.2020.9235263. — URL: http://dx.doi.org/10.1109/MFI49285.2020.9235263.

McInnes, *L.* hdbscan: Hierarchical density based clustering / L. McInnes, J. Healy, S. Astels // The Journal of Open Source Software. — 2017. — Mar. — Vol. 2, no. 11. — DOI: 10.21105/joss.00205. — URL: https://doi.org/10.21105%2Fjoss.00205.

McInnes, *L.* Umap: Uniform manifold approximation and projection for dimension reduction / L. McInnes, J. Healy, J. Melville // arXiv preprint arXiv:1802.03426. — 2018.

McInnes, *L.* UMAP: Uniform Manifold Approximation and Projection / L. McInnes, J. Healy, N. Saul, [et al.] // The Journal of Open Source Software. — 2018. — Vol. 3, no. 29. — P. 861.

Muennighoff, *N.* MTEB: Massive Text Embedding Benchmark / N. Muennighoff [et al.]. — 2023. — arXiv: 2210.07316 [cs.CL]. — URL: https://arxiv.org/abs/2210.07316.

Nomura, *M.* CMA-ES with learning rate adaptation: Can CMA-ES with default population size solve multimodal and noisy problems? / M. Nomura, Y. Akimoto, I. Ono // Proceedings of the Genetic and Evolutionary Computation Conference. — 2023. — P. 839–847.

Nomura, *M.* cmaes: A simple yet practical python library for cma-es / M. Nomura, M. Shibata // arXiv preprint arXiv:2402.01373. — 2024.

Nomura, *M.* Warm starting CMA-ES for hyperparameter optimization / M. Nomura, S. Watanabe, [et al.] // Proceedings of the AAAI conference on artificial intelligence. Vol. 35. — 2021. — P. 9188–9196.

Nothman, *J.* Stop word lists in free open-source software packages / J. Nothman, H. Qin, R. Yurchak // Proceedings of workshop for NLP open source software (NLP-OSS). — 2018. — P. 7–12.

Nussbaum, *Z.* Nomic Embed: Training a Reproducible Long Context Text Embedder / Z. Nussbaum [et al.]. — 2024. — arXiv: 2402.01613 [cs.CL].

Ortega, *J. D.* Multimodal fusion with deep neural networks for audio-video emotion recognition / J. D. Ortega [et al.] // arXiv preprint arXiv:1907.03196. — 2019.

Pathak, *A. R.* Application of Deep Learning Approaches for Sentiment Analysis / A. R. Pathak [et al.] // Deep Learning-Based Approaches for Sentiment Analysis / ed. by B. Agarwal [et al.]. — Singapore : Springer Singapore, 2020. — P. 1–31. — ISBN 978-981-15-1216-2. — DOI: 10.1007/978-981-15-1216-2_1. — URL: https://doi.org/10.1007/978-981-15-1216-2_1.

Rajpurkar, *P.* Squad: 100,000+ questions for machine comprehension of text / P. Rajpurkar [et al.] // arXiv preprint arXiv:1606.05250. — 2016.

Raschka, *S.* Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence / S. Raschka, J. Patterson, C. Nolet // arXiv preprint arXiv:2002.04803. — 2020.

Reimers, *N.* Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks / N. Reimers, I. Gurevych // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. — Association for Computational Linguistics, 11/2019. — URL: https://arxiv.org/abs/1908.10084.

Sainburg, *T.* Parametric UMAP: learning embeddings with deep neural networks for representation and semi-supervised learning / T. Sainburg, L. McInnes, T. Q. Gentner // ArXiv e-prints. — 2020. — arXiv: 2009.12981 [stat.ML].

Shah, *R. S.* When FLUE Meets FLANG: Benchmarks and Large Pretrained Language Model for Financial Domain / R. S. Shah [et al.] // Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP). — Association for Computational Linguistics, 2022.

Shahriari, *B.* Taking the human out of the loop: A review of Bayesian optimization / B. Shahriari [et al.] // Proceedings of the IEEE. — 2015. — Vol. 104, no. 1. — P. 148–175.

Shazeer, *N.* Outrageously large neural networks: The sparsely-gated mixture-of-experts layer / N. Shazeer [et al.] // arXiv preprint arXiv:1701.06538. — 2017.

Sinha, *A.* Impact of news on the commodity market: Dataset and results / A. Sinha, T. Khandait // Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2. — Springer. 2021. — P. 589–601.

Socher, *R.* Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank / R. Socher [et al.] // Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. — Seattle, Washington, USA : Association for Computational Linguistics, 10/2013. — P. 1631–1642. — URL: https://www.aclweb.org/anthology/D13-1170.

Vanstone, *B. J.* Do news and sentiment play a role in stock price prediction? / B. J. Vanstone, A. Gepp, G. Harris // Applied Intelligence. — 2019. — Nov. — Vol. 49, issue 11. — P. 3815–3820. — ISSN 15737497. — DOI: 10.1007/s10489-019-01458-9.

Vaswani, *A.* Attention is all you need / A. Vaswani [et al.] // Advances in neural information processing systems. — 2017. — Vol. 30.

Vijayan, *D.* Adaptive hierarchical density-based spatial clustering algorithm for streaming applications / D. Vijayan, I. Aziz // Telecom. Vol. 4. — MDPI. 2022. — P. 1–14.

Vuković, *D. B.* Predictive Patterns and Market Efficiency: A Deep Learning Approach to Financial Time Series Forecasting / D. B. Vuković [et al.] // Mathematics. — 2024. — Oct. — Vol. 12, issue 19. — ISSN 22277390. — DOI: 10.3390/math12193066.

Wang, *A.* GLUE: A multi-task benchmark and analysis platform for natural language understanding / A. Wang [et al.] // arXiv preprint arXiv:1804.07461. — 2018.

Wang, *Y.* Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization / Y. Wang [et al.] // Journal of Machine Learning Research. — 2021. — Vol. 22, no. 201. — P. 1–73. — URL: http://jmlr.org/papers/v22/20-1061.html.

Warner, *B.* Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference / B. Warner [et al.]. — 2024. — Dec. — URL: http://arxiv.org/abs/2412.13663.

Watanabe, *S.* Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance / S. Watanabe // arXiv preprint arXiv:2304.11127. — 2023.

Wen, *Q.* Transformers in time series: A survey / Q. Wen [et al.] // arXiv preprint arXiv:2202.07125. — 2022.

Wiebe, *J.* Annotating expressions of opinions and emotions in language / J. Wiebe, T. Wilson, C. Cardie // Language resources and evaluation. — 2005. — Vol. 39. — P. 165–210.

Williams, *A.* A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference / A. Williams, N. Nangia, S. R. Bowman. — 2018. — arXiv: 1704.05426 [cs.CL]. — URL: https://arxiv.org/abs/1704.05426.

Wu, *S.* Bloomberggpt: A large language model for finance / S. Wu [et al.] // arXiv preprint arXiv:2303.17564. — 2023.

Xing, *F.* Financial sentiment analysis: An investigation into common mistakes and silver bullets / F. Xing [et al.] // Proceedings of the 28th international conference on computational linguistics. — 2020. — P. 978–987.

Yang, *Y.* FinBERT: A Pretrained Language Model for Financial Communications / Y. Yang, M. C. S. UY, A. Huang. — 2020. — June. — URL: http://arxiv.org/abs/2006.08097.

Zhang, *X.* mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval / X. Zhang [et al.] // Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track. — 2024. — P. 1393–1412.