Predicting Market Reactions to News:

An LLM-Based Approach Using Spanish Business Articles

Jesus Villota *

Abstract

Markets do not always efficiently incorporate news, particularly when information is complex or ambiguous. Traditional text analysis methods fail to capture the economic structure of information and its firm-specific implications. We propose a novel methodology that guides LLMs to systematically identify and classify firm-specific economic shocks in news articles according to their type, magnitude, and direction. This economically-informed classification allows for a more nuanced understanding of how markets process complex information. Using a simple trading strategy, we demonstrate that our LLM-based classification significantly outperforms a benchmark based on clustering vector embeddings, generating consistent profits out-of-sample while maintaining transparent and durable trading signals. The results suggest that LLMs, when properly guided by economic frameworks, can effectively identify persistent patterns in how markets react to different types of firm-specific news. Our findings contribute to understanding market efficiency and information processing, while offering a promising new tool for analyzing financial narratives.

JEL Codes: G12, G14, C45, C58, C63, D83

Keywords: Large Language Models, Business News, Stock Market Reaction, Market Efficiency

* CEMFI, Calle Casado del Alisal, 5, 28014, Madrid, Spain. I am deeply grateful to Enrique Sentana, Manuel Arellano, Dante Amengual, Rafael Repullo, Javier Suárez, David Martínez-Miera, Julio Crego, and Francisco Peñaranda for their comments. I am especially indebted to Nicolás Forteza and Matías Covarrubias, whose invaluable guidance and insights have greatly enriched this work. I also thank discussants for their helpful comments, as well as the participants of the Banking & Finance Seminar (CEMFI), the Generative AI in Finance Conference (Concordia University), the Mirian Andrés Seminar (University of La Rioja), the 3rd Contemporary Issues in Financial Markets and Banking (Nottingham Trent University), the Sao Paulo School of Advanced Science in High Dimensional Models (FGV) and the Barcelona Summer Forum: Machine Learning in Economics (BSE). Finally, I gratefully acknowledge financial support from Banco de España (BdE).

1. Introduction

In financial markets, news play a pivotal role in shaping stock prices. Every day, market participants respond to a broad spectrum of news ranging from firm-specific announcements, such as earnings releases, to macroeconomic events, such as central bank interest rate announcements, or geopolitical developments, like international trade conflicts or political elections. The Efficient Market Hypothesis (EMH), formalized by [1] posits that markets efficiently incorporate new information almost instantaneously. Both theoretical perspectives and empirical observations indicate that markets do not always exhibit such efficiency, particularly when the information is complex or ambiguous. This discrepancy between theory and reality suggests significant room for improvement in understanding how news is processed by market participants and how it influences asset prices. A substantial body of literature has tried to predict market reactions to news, yet some important gaps persist. Our review of the literature reveals three critical limitations in current approaches to analyzing financial news: a lack of economic focus in textual analysis methodology, insufficient attention to firm-specific effects, and over-reliance on headlines.

First, we examine the lack of economic focus in current methodological approaches to analyzing financial news. This limitation is evident across three main streams of literature.

Sentiment Analysis. Traditional approaches frequently rely on sentiment analysis, reducing the richness of news content to binary classifications of positive or negative sentiment. The seminal work of [2] demonstrated the predictive power of media sentiment in financial markets, showing that negative media coverage leads to downward pressure on market prices, followed by a reversion to fundamentals. This finding sparked significant interest in sentiment-based approaches, with [3] extending the analysis to firm-specific news and revealing that negative word content not only forecasts poor firm earnings but also indicates a temporary underreaction in stock prices. Despite these early successes, the methodology of sentiment analysis has faced important challenges. [4] highlighted a fundamental issue: general-purpose dictionaries often misclassify words in financial contexts, leading them to develop specialized financial word lists. Building on this insight, [5] demonstrated that the weighting scheme applied to these words is as crucial as the word lists themselves, introducing a more nuanced approach to content analysis. The emergence of social media and machine learning has driven further methodological innovations in sentiment analysis. [6] leveraged Twitter data to predict DJIA movements, while [7] revealed that sentiment's predictive power is particularly pronounced during recessions, suggesting time-varying importance of news sentiment. Recent advances in machine learning have pushed the boundaries further, with [8] developing a sophisticated supervised learning framework specifically designed for return prediction. The advent of transformer-based models has enabled even more sophisticated approaches, with [9] and [10] applying BERT-based architectures to financial sentiment analysis. However, despite

their widespread adoption and continued methodological refinements, sentiment analysis approaches remain fundamentally limited. They often miss the intricacy inherent in news by focusing on linguistic patterns rather than economically relevant considerations.

Topic modeling. Beyond sentiment analysis, researchers have also explored topic modeling as an alternative approach to categorize text into broader themes. The pioneering work of [11] demonstrated that computational linguistics methods could reveal important patterns in market reactions to news, finding that stock prices do not immediately and consistently reflect news, with effects varying significantly across different types of stories and market conditions. Topic modeling approaches have since been applied across financial research domains. [12] used these techniques to analyze Federal Reserve communications, while [13] developed a topic model analyzing over 800,000 Wall Street Journal articles to track news attention to different economic themes. [14] further integrated topic modeling with asset pricing models to derive systematic risk factors from news. However, these models are limited in adapting to new and evolving information and lack the specificity needed to assess the precise impact of news on individual firms or sectors. While topic models can identify broad themes, they struggle to capture the changing context of financial news, particularly when new narratives emerge, such as unexpected geopolitical events or technological disruptions.

Vector-based models. Vector-based models have emerged as an alternative approach to address the limitations of both sentiment analysis and topic modeling. The foundational models in this domain, Word2Vec and GloVe, established the paradigm of mapping words to continuous vector spaces based on their co-occurrence patterns, enabling mathematical operations on words and capturing semantic relationships. [15] pioneered their application in finance by developing time-varying measures of product similarity from firms' 10-K descriptions, demonstrating how vector representations could capture nuanced competitive relationships that traditional industry classifications miss. The advent of transformer architectures marked a significant advancement, leading to more sophisticated models such as BERT, RoBERTa, or GPT. These models process text through multiple attention layers, generating context-aware embeddings by considering relationships between all words simultaneously. [16] demonstrated their superior performance in predicting stock movements following financial news events, while [17] leveraged BERT to develop a novel measure of bond "greenness", revealing how subtle textual differences in bond documentation translate into measurable price effects. Recent applications have further expanded the scope of these methods. [18] analyzed finance sentiment across multiple countries and centuries, while [19] integrated sentiment analysis from GPT and BERT into traditional asset pricing models. [20] introduced "*asset embeddings*", showing how these techniques can uncover latent firm characteristics from investors' holdings data. However, even when fine-tuned with domain-specific training data (e.g: `FinBERT`), these methods

cannot inherently incorporate economic structure, which limits their ability to comprehend the economic implications of news articles.

Having examined the limitations of current methodological approaches, we now turn to a second critical gap in the literature: there is an insufficient focus on firm-specific analysis in existing research. Many studies examine the impact of news on broader market indices such as the S&P500 or DJIA, rather than on individual firms. For example, [21] and [22] analyzed comprehensive news coverage to understand aggregate market movements, while more recent work has leveraged increasingly sophisticated data sources. [6] developed novel mood tracking tools for Twitter messages to predict DJIA movements, and [7] examined a century of New York Times financial columns to study market-wide returns during recessions. [23], [24] and [25] constructed innovative news-based indices that have enhanced our understanding of market-wide uncertainty and volatility. While these and other similar studies provide valuable insights into market-wide reactions, they fall short in elucidating how specific firms are affected by news events. Firm-specific impacts are often masked when aggregated at the index level, leading to a loss of critical information about how particular entities are influenced by specific news. For example, during the COVID-19 pandemic, market indices masked substantial heterogeneity in firm-level responses with some sectors like technology and healthcare experiencing positive returns, while others, such as hospitality, travel, and retail, experiencing significant negative impacts due to widespread lockdowns and reduced consumer spending. Such differences are often obscured when focusing solely on market indices. Tools like Named Entity Recognition (NER), which could help identify firms impacted by particular events, remain underutilized in financial research, further contributing to the lack of firm-level granularity.

The third and final critical issue is the over-reliance on headlines as the basis for news analysis. Headlines are often used due to their availability and the simplicity of extracting sentiment from them, making them convenient but insufficient for comprehensive analysis. [26] provided early evidence of this limitation, showing distinct market reactions to headline news versus no-news events, particularly in terms of drift after bad news and reversals after extreme price movements. As natural language processing techniques evolved, researchers continued to focus primarily on headlines: [27] and [10] applied increasingly sophisticated deep learning and BERT models to headline analysis, while recent work by [28] and [29] has extended this approach using large language models to extract contextualized representations from news headlines. While these studies have advanced our understanding of market reactions to news, headlines are designed to capture attention rather than provide comprehensive information. Consequently, relying solely on headlines can lead to overly simplistic analyses that fail to capture critical contextual details necessary for accurately predicting market reactions.

This paper seeks to address these three limitations by leveraging Large Language Models (LLMs) to facilitate an economically-structured, granular and firm-specific analysis of complete news articles. LLMs are particularly suited for economic interpretation due to their extensive training on human-generated

text, including financial and economic discourse. This exposure enables them to "*understand*" economic concepts, cause-and-effect relationships, and market mechanisms in ways that mirror human economic reasoning. Unlike purely statistical approaches, LLMs can recognize economic patterns and implications that would be evident to market participants, making them powerful tools for financial analysis. For example, LLMs could simulate human analysis of news articles, understanding the economic shocks that a news article describes upon a specific firm –such as supply chain disruptions affecting manufacturing, shifts in consumer demand impacting retail, or policy changes influencing energy sectors– and quantifying both the magnitude and direction of these impacts on specific firms. In this study, we leverage LLMs to parse a dataset of Spanish business news articles from DowJones Newswires, spanning June 2020 to September 2021, a particularly unstable period marked by economic disruptions due to the COVID-19 pandemic. This period was purposefully chosen for its inherent complexity and market instability. Testing our methodology during such a challenging period allows us to rigorously evaluate its robustness and effectiveness. While many methodologies can perform adequately during stable market conditions, their true capabilities are revealed when faced with unprecedented market dynamics and rapid economic changes.

Our methodology consists of defining a schema with which we guide an LLM to detect firm-specific shocks from business news and to further classify them by their type (demand, supply, technological, policy, financial), magnitude (minor, major) and direction (positive, negative). Through their ability to categorize and comprehend the economic implications of news, LLMs generate insights that surpass traditional methodologies, revealing the underlying mechanisms driving market behavior. This allows for a more detailed assessment of how specific pieces of information influence particular firms, providing a richer and more precise picture of market dynamics. As our benchmark, we employ a vector-based approach that represents each news article as a high-dimensional embedding vector using a sentence transformer. This benchmark choice serves two key purposes. First, it offers greater granularity and sophistication compared to traditional methods like sentiment analysis and topic modeling. Second, it provides theoretical consistency with our LLM-based approach, as vector embeddings constitute the first layer of an LLM's architecture. This parallel allows us to effectively compare the predictive power of the LLM's initial representation (vector embeddings) with its final output (economically structured news classification). Through this comparison, we can assess whether incorporating economic structure in the LLM processing step enhances our ability to predict market reactions to news.

To evaluate the timing ability of our proposed methodology, we develop a trading strategy that builds on the traditional portfolio sorting approach. While conventional strategies sort stocks based on firm characteristics, we instead sort based on news clusters. For the benchmark (vector embeddings), we employ KMeans clustering, while our LLM methodology clusters articles by shock categories. We identify the best and worst-performing clusters by analyzing the stock price responses of affected firms, then construct a long-short portfolio strategy that takes long positions in the best-performing clusters and

short positions in the worst-performing ones. The profitability of this strategy serves as a measure of each clustering methodology's ability to identify economically meaningful news patterns that translate into improved market timing abilities. Our findings reveal that while the vector-based model successfully identifies firm- and industry-specific clusters, its trading signals lack persistence. The model's reliance on historical firm and industry performance patterns generates ephemeral signals that do not translate well to future market conditions. In contrast, our LLM-based methodology produces clusters based on economically meaningful shock classifications, resulting in more persistent trading signals. The superior out-of-sample performance of our LLM-based trading strategy demonstrates its enhanced capability to capture and interpret market reactions to news, underscoring the advantages of incorporating economic structure into news analysis.

The objective of this paper is not to parse the largest dataset available or to develop a realistic trading strategy with commercial application. Rather, it aims to introduce a novel methodology for analyzing news articles in a granular and firm-specific manner, demonstrating its utility through a reduced dataset. By focusing on a smaller, high-quality dataset, the study emphasizes methodological rigor and interpretability. The findings are intended to contribute to a more nuanced understanding of how market participants process news, using a simple trading strategy to illustrate the potential of this approach in capturing the complexities of information processing in financial markets. This methodological contribution lays the groundwork for future research that could extend these techniques to larger datasets and more complex trading applications, ultimately enhancing our ability to understand and predict market behavior in response to news.

The remainder of this paper is organized as follows: Section 2 presents the dataset and preprocessing steps. Section 3 provides a mathematical framework for analyzing news articles. In Section 4, we focus on clustering news articles – first presenting the benchmark framework using KMeans clustering of vector embeddings, followed by our novel LLM-based methodology. Section 5 details the construction of a simple trading strategy, including market-beta-neutral positions for each firm-article pair, extraction of cluster-average Sharpe Ratios, and selection of optimal clusters based on two proposed algorithms. In Section 6, we perform robustness checks by examining the sensitivity of our results to hyperparameter variations. Finally, Section 7 concludes and discusses the implications of our findings

2. Data

This paper employs a dataset of Spanish business news articles sourced from Dow Jones Newswires, covering the period from June 24, 2020, to September 30, 2021. The selection of this timeframe is deliberate, driven by two key considerations. First, given the substantial computational demands of LLM-based analysis, we strategically focus on a smaller, carefully curated dataset. This deliberate scope reduction allows us to thoroughly demonstrate our novel methodology's effectiveness in decoding market-

news relationships while keeping computational costs manageable. Second, we specifically chose the Covid-19 era to test our methodology's extrapolative capabilities during periods of significant market instability and volatility. While existing textual algorithms typically perform well in stable market conditions, they often struggle to generalize effectively during periods of heightened uncertainty. By focusing on this volatile period, we can better assess our methodology's robustness and its ability to maintain predictive power under challenging market conditions.

The dataset consists of high-quality articles that have been filtered to include only those mentioning Spanish publicly traded firms listed on the IBEX-35 index. These 35 companies represent the largest firms in Spain by market capitalization and are typically the most liquid and actively traded Spanish stocks. Moreover, these companies tend to receive the most consistent media coverage, making them ideal for the scope of our analysis.

The use of Dow Jones Newswires as our news source is also intentional. Dow Jones has a standard practice of including the stock market ticker of firms directly affected by the article in parentheses, while excluding firms mentioned for secondary purposes from ticker specification. This feature significantly facilitates the extraction of named entities (i.e., Named Entity Recognition, or NER). The tickers used by Dow Jones align with those from Yahoo Finance, enabling seamless integration between our NER algorithm and subsequent firm-specific trading operations via the Yahoo Finance API. We employ a pattern recognition algorithm through the `regex` library in Python to identify specific mentions of publicly traded companies in the Spanish stock exchange. The algorithm searches for patterns of the form "`(<WORD>.MC)`" for any `<WORD>`. For instance, consider the following example article (translated into English for convenience):

Example 1: An article about ACS and Acciona (translated into English)

ACS and Acciona Secure Contracts for New Australian Airport

A consortium of Actividades de Construcción y Servicios SA (ACS.MC) and Acciona SA (ANA.MC) has won a contract to build the operations area of the Western Sydney International Airport (Nancy-Bird Walton) and carry out paving works, amounting to AUD265 million (EUR164 million) for the Australian subsidiary CIMIC Group Ltd (CIM.AU). CIMIC will carry out the work through its subsidiary CPB Contractors, as stated in a press release. This is the third project awarded by Western Sydney Airport to the joint venture after being selected to carry out earthworks. Construction will take two years, and the Western Sydney airport is expected to open in 2026.

Our NER algorithm applied to Example 1 successfully identifies the Spanish firms `ACS.MC` (Actividades de Construcción y Servicios SA) and `ANA.MC` (Acciona SA) while disregarding the Australian `CIM.AU` (CIMIC Groups Ltd). To further ensure the reliability of firm identification, we validate the extracted entities using a Large Language Model (LLM). In particular, we feed the articles to the LLM, which parses

them according to a predefined schema. As we will see later, the first task in this schema is to identify the listed Spanish firms directly affected by the events described in the article. Finally, the identified firms are filtered against a dynamic list of IBEX-35 members. Due to the high quality of the dataset, the correlation between entities identified by the LLM and those extracted via pattern recognition is almost exact.

For subsequent analysis, we partition the dataset into three splits: *Train*, *Validation*, and *Test*. Each split serves a distinct purpose that will be explained in detail as we progress through the paper. Summary statistics for each data split are provided in Table 1.

[Insert Table 1 about here]

The most frequently used words in the whole dataset are depicted in Figure 1 by means of a WordCloud. As shown, the most prominent words include "*empresa*" (firm), "*compañía*" (company), and "*españa*" (Spain), reinforcing that the dataset primarily comprises Spanish business news, with a prevalence of technical terms such as "*beneficio neto*" (net profit), "*precio objetivo*" (target price), "*proyecto*" (project), and "*operación*" (operation).

[Insert Figure 1 about here]

The distribution of the number of articles published per day is illustrated in Figure 2a, showing that the most frequent publication rate is between 5 and 10 articles per day, though some days exhibit unusually high publication counts. Figure 2b shows the distribution of the number of words per article, with the majority of articles containing between 70 and 280 words. This indicates that the articles are relatively succinct, providing direct information. However, the long right tail points to instances of more comprehensive coverage.

[Insert Figure 2 about here]

The time series of the number of articles published per day throughout the sample period is shown in Figure 3. The series exhibits considerable variability, with frequent fluctuations from fewer than 5 articles per day to sudden spikes exceeding 20 articles. The 30-day moving average smooths the series, confirming the previous observation that, on average, between 5 and 10 articles are published daily.

[Insert Figure 3 about here]

Data Availability. The dataset used in this study contains confidential information provided under agreements with the Bank of Spain and Dow Jones Newswires, and cannot be shared publicly or with third parties. Interested readers may access the same data from Dow Jones Newswires for a fee.

3. Mathematical Treatment of News Articles

Our dataset consists of $N = 2,613$ Spanish business news articles sourced from DowJones and spanning the period from 2020/06/24 to 2021/09/30. We denote as $\mathcal{D}$ the set of all articles in our sample. These articles have been specifically filtered to reference firms listed on the IBEX-35. Let $\mathcal{F}_{\text{IBEX35}}$ denote the universe of such firms. Each article $i \in \mathcal{D}$ is a textual document detailing an event that directly pertains to a subset of firms $\mathcal{F}^i \subseteq \mathcal{F}_{\text{IBEX35}}$. The publication date and time of each article are represented as $\langle d_0^i, t_0^i \rangle$, where d_0^i captures the date (YYYY-MM-DD) and t_0^i captures the time (HH:MM) of publication. Therefore we observe the moment at which $\mathcal{F}^i$ receives the "*treatment*" of public news dissemination.

Effective treatment day

We are interested in examining the impact of each news article $i \in \mathcal{D}$ on the stock price of the firms directly affected by it (i.e., all $j \in \mathcal{F}^i$). Since publication datetime may not coincide with trading hours, we define an *effective treatment date*, denoted $\tilde{d}_0^i$. This maps the news article publication datetime to the nearest trading date where the stock price can reflect the news impact.

Let $\mathfrak{d}$ denote the set of all dates in our sample timeline and let $\tilde{\mathfrak{d}} \subset \mathfrak{d}$ denote the subset of Spanish trading days. We define a function $\Lambda : \mathfrak{d} \to \tilde{\mathfrak{d}}$ that finds the next trading date after a given date: $\Lambda(d) := \min\{\tilde{d} \in \tilde{\mathfrak{d}} \mid \tilde{d} > d\}$. We set $\tilde{d}_0^i$ to the publication date if the article was published on a trading day before market close (17:30 in Spain), and to the next trading day otherwise. Formally,

$$\tilde{d}_0^i := \begin{cases} d_0^i & \text{if} \quad d_0^i \in \tilde{\mathfrak{d}} \ \wedge \ t_0^i < 17{:}30 \\ \Lambda(d_0^i) & \text{if} \quad d_0^i \notin \tilde{\mathfrak{d}} \ \vee \ t_0^i \geq 17{:}30 \end{cases}.$$

The two possible cases are illustrated in Figure 4.

[Insert Figure 4 about here]

Data Splitting

For robust model development and evaluation, the dataset is partitioned into three sequential subsets: training, validation, and test: $\mathcal{D} := \mathcal{D}^{tr} \cup \mathcal{D}^{val} \cup \mathcal{D}^{test}$. Define $N_{split} := |\mathcal{D}^{split}|$ for $split \in \{tr, val, test\}$, where $|\cdot|$ denotes the cardinality of a set. The training and validation sets collectively comprise 80% of the total dataset ($\frac{N_{\text{tr}}+N_{\text{val}}}{N} = 0.8$) and are instrumental in constructing and fine-tuning the trading strategy. The remaining 20% ($\frac{N_{\text{test}}}{N} = 0.2$) is reserved for out-of-sample testing to assess the performance and generalizability of the strategy under unseen conditions.

4. Clustering News Articles

In this section we present our clustering methodology based on news-implied firm-specific shock classifications and we compare it against a benchmark based on clustering the vector embedding representations of the articles. For ease of exposition, we will first present the benchmark model.

4.1 Benchmark: KMeans clustering of vector embeddings

4.1.1 Why this benchmark?

In evaluating our novel Large Language Model (LLM) methodology for classifying news-implied firm-specific shocks, we selected KMeans clustering of high-dimensional vector embeddings as the benchmark over alternatives like sentiment analysis and topic modeling. Sentiment analysis, while straightforward, lacks the necessary granularity, offering only positive, negative, or neutral classifications, which is insufficient to compare with our granular LLM-based economic shock classification. Additionally, sentiment analysis focuses on the emotional tone rather than the economic impact, it is prone to inconsistencies due to linguistic nuances and it can deliver very different outcomes depending on the specific sentiment analysis tool employed.

On the other hand, topic modeling provides more detailed classifications than sentiment analysis but relies on bag-of-words representations that fail to capture complex semantic relationships and contextual nuances essential for identifying economic shocks accurately. Vector embeddings, particularly those generated by transformer-based models, offer enhanced semantic representation by capturing context-dependent meanings and scaling efficiently with large datasets, making them more flexible and adaptable for clustering and classification. Although embeddings lack inherent interpretability, this issue is addressed by clustering, which allows us to infer meaningful firm-specific or industry-specific patterns from the grouped articles.

Lastly, using embeddings as a benchmark is particularly compelling because they represent the foundational layer of an LLM. Namely, the first step in an LLM's processing pipeline is to transform the text that it is fed into embeddings for further processing. By benchmarking against embeddings, we ensure a direct and relevant comparison between the foundational representations used by LLMs and our specialized classification methodology. This comparison highlights the added value of the LLM's capacity to convert these semantic representations (i.e: the vector embeddings) into economically meaningful classifications. (i.e: our news-implied firm-specific shock classifications). Consequently, KMeans clustering of vector embeddings provides a robust, scalable, and economically pertinent benchmark, superior to sentiment analysis and topic modeling, for assessing our LLM-based classification of news-implied firm-specific shocks. A more detailed discussion can be found in A.7.

4.1.2 Vector embeddings: "*Transforming text into high-dimensional vectors*"

Any piece of text can be represented as a high-dimensional vector embedding by using a transformer. Transformers are a type of deep learning architecture introduced by [30] which have revolutionized natural language processing (`NLP`). The core idea behind them is the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence when generating a representation for each word. This mechanism enables transformers to capture long-range dependencies and contextual relationships within the text more effectively than previous models like recurrent neural networks (`RNN`s).

A transformer model consists of an encoder (and potentially, a decoder as well) composed of multiple layers of self-attention and feedforward neural networks. In our context, we primarily use the encoder to convert a piece of text into a fixed-size vector, known as an embedding. Since our articles are written in Spanish, we employ a `Multilingual Sentence Transformer`, which has been trained on text from multiple languages.

For every news article $i \in \mathcal{D}$, we obtain a representative vector embedding $\mathbf{e}^i \in \mathbb{R}^{512}$ that provides a numerical representation of various aspects of the text, such as syntactic structure, semantic content, and contextual nuances. While it is challenging to assign a specific human-readable meaning to each of the 512 components, we can interpret the vector as a whole in various ways:

- *Semantic Similarity*: Similar articles will have similar embeddings. For instance, if one article discusses a company's quarterly earnings and another article discusses the same company's annual earnings, their embeddings will be close in the 512-dimensional space.
- *Topic Clustering*: Articles on similar topics will cluster together. For example, articles about financial markets might cluster in one region of the embedding space, while articles about mergers and acquisitions cluster in another.
- *Sentiment Analysis*: Different regions of the embedding space can implicitly represent different sentiments. Articles with positive news might cluster in one area, while those with negative news cluster in another.

4.1.3 Clustering embeddings with KMeans

With the numerical representation of each article in the form of embeddings $\{\mathbf{e}^i\}_{i\in\mathcal{D}}$, we now seek to identify groups of similar articles. Namely, we use the KMeans algorithm, a popular clustering method that assigns a set of vectors into k clusters $\mathcal{G}_{\text{KMeans}} := \{0, 1, ..., k-1\}$ to minimize the within-cluster sum of squares (WCSS). The implementation of this clustering algorithm is methodically presented in Appendix Algorithm 1. Each cluster $g \in \mathcal{G}_{\text{KMeans}}$ defines a centroid $\mathbf{c}_g$, which is the average vector of all

the members of a cluster. In the first step, we apply the algorithm to the training data ($\mathcal{D}^{tr}$).

$$\begin{aligned} \min_{\{\mathcal{D}_g^{tr}\},\{\mathbf{c}_g\}} \quad & \textstyle\sum_{g=1}^{k} \sum_{i\in\mathcal{D}_g^{tr}} \|\mathbf{e}^i - \mathbf{c}_g\|_2^2 \\ \text{s.t.} \quad & \left| \begin{array}{l} \bigcup_{g=1}^{k} \mathcal{D}_g^{tr} = \mathcal{D}^{tr} \\ \mathcal{D}_g^{tr} \cap \mathcal{D}_h^{tr} = \emptyset \quad \forall g,h \in \mathcal{G}_{\text{KMeans}} : g \neq h \end{array} \right. . \end{aligned}$$

The optimal number of clusters k^* in this algorithm is to be set exogenously. Here, we take it to maximize the average silhouette score in the training sample over some grid $\mathbf{k}$ of cluster sizes k:

$$k^* := \arg\max_{k\in\mathbf{k}} \frac{1}{|\mathcal{D}^{tr}|} \sum_{i\in\mathcal{D}^{tr}} s_k(\mathbf{e}^i) \ .$$

The silhouette score $s_k(\mathbf{e}^i) \in [-1,1]$ measures how well an embedding is clustered by comparing its similarity to its own cluster (*intra-cluster distance*) with its similarity to the nearest other cluster (*inter-cluster distance*). A clustering configuration with a higher average silhouette score (close to +1) is considered better because it indicates that clusters are dense and well-separated. Formally, the silhouette score is defined as

$$s_k(\mathbf{e}^i) := \frac{b_k(\mathbf{e}^i) - a_k(\mathbf{e}^i)}{\max\{a_k(\mathbf{e}^i), b_k(\mathbf{e}^i)\}} \ ,$$

where, for $i \in \mathcal{D}_g^{tr}$, the *intra-cluster distance* is defined as $a_k(\mathbf{e}^i) := (|\mathcal{D}_g^{tr}| - 1)^{-1} \sum_{m\in\mathcal{D}_g^{tr}, m\neq i} \|\mathbf{e}^i - \mathbf{e}^m\|_2$ and it represents the average distance from an embedding $\mathbf{e}^i$ to all other embeddings in the same cluster, while the *inter-cluster distance* is $b_k(\mathbf{e}^i) := \min_{l\neq g}(|\mathcal{D}_l^{tr}|)^{-1} \sum_{m\in\mathcal{D}_l^{tr}} \|\mathbf{e}^i - \mathbf{e}^m\|_2$ and it represents the minimum average distance from an embedding $\mathbf{e}^i$ to all embeddings in the nearest different cluster.

In Figure 5 we plot the average silhouette score for $\mathcal{D}^{tr}$ computed over a grid $\mathbf{k}$ ranging from 2 to 100. The vertical dashed green line signals the maximizer of the grid, which corresponds to a cluster size of $k^* = 26$.

[Insert Figure 5 about here]

Given the optimal number of clusters k^*, we fit the KMeans algorithm on the training embeddings $\{\mathbf{e}^i \mid i \in \mathcal{D}^{tr}\}$ to obtain the centroids $\{\mathbf{c}_1^{tr}, \mathbf{c}_2^{tr}, \ldots, \mathbf{c}_{k^*}^{tr}\}$. Following Algorithm 1. (detailed in A.1):

$$\{\mathbf{c}_1^{tr}, \mathbf{c}_2^{tr}, \ldots, \mathbf{c}_{k^*}^{tr}\} = \text{KMeans}(\{\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^{N_{tr}}\}, k^*) \ .$$

We then find the cluster associated to each embedding $\mathbf{e}^i$ in the validation set $\{\mathbf{e}^i \mid i \in \mathcal{D}^{val}\}$ according to the centroids resulting from clustering the training data $\{\mathbf{c}_1^{tr}, ..., \mathbf{c}_{k^*}^{tr}\}$. This allows us to obtain the clustering of the news articles in the validation sample

$$\mathcal{D}_g^{val} = \left\{ i \in \mathcal{D}^{val} \;\middle|\; g = \arg\min_{\ell\in\mathcal{G}} \|\mathbf{e}^i - \mathbf{c}_\ell^{tr}\|_2^2 \right\} \quad \forall g \in \mathcal{G}_{\text{KMeans}}.$$

Similarly, by assigning each embedding $\mathbf{e}^i \in \{\mathbf{e}^i \mid i \in \mathcal{D}^{test}\}$ to the nearest centroid $\mathbf{c}_g^{tr}$, we obtain the clusters in the test set

$$\mathcal{D}_g^{test} = \left\{ i \in \mathcal{D}^{test} \;\middle|\; g = \arg\min_{\ell \in \mathcal{G}} \|\mathbf{e}^i - \mathbf{c}_\ell^{tr}\|_2^2 \right\} \quad \forall g \in \mathcal{G}_{\text{KMeans}}.$$

[Insert Figure 6 about here]

In Figure 6 we can see that the distribution of articles in the whole sample ($\mathcal{D}$) is fairly homogenous across the 26 clusters, with each cluster containing between 50 and 250 articles on average. The notable exceptions are cluster 3, which contains only 24 articles, and cluster 4, which concentrates 428 articles. However, the distribution profile is not consistent over data splits, which indicates that this classification procedure is unstable over time.

Although not directly interpretable, by looking at the articles pooled in a certain cluster, we can provide some intuition of what it represents. In most cases, each cluster contains articles involving a firm or set of firms in the same sector. For example, cluster 3 pools articles about Telefónica and Cellnex (telecoms), cluster 4 contains articles about CaixaBank, cluster 9 concentrates articles about Repsol, cluster 12 about Iberdrola, cluster 15 gathers articles on Infrastructure (led by ACS and Acciona) and so on.

However, there are some exceptions to this general rule, for example, cluster 0 is a "*miscellanous*" cluster: it covers articles about different firms with no apparent relation between them. Another example is cluster 1, which pools articles related to the quarterly or semiannual publication of results by different firms. In Appendix Table A1 we provide a sample of 3 articles for each cluster and propose a name for each one based on the articles they pool.

4.2 LLM-based approach: "*What if an LLM reads the news?*"

One may wonder whether empowering an LLM to parse news articles according to a predefined schema that guides it in ellucidating news-implied firm-specific shocks can deliver better insights on how markets react to new information. In this section we will briefly introduce what Large Language Models are, how they have evolved and then, we will dive into how we can guide them to produce an economically structured analysis of business news.

4.2.1 Large Language Models

In natural language processing (NLP), Large Language Models (LLMs) are designed to "*understand*" and generate human-like text. These models utilize the transformer architecture, which excels in modeling complex language tasks by capturing long-range dependencies and contextual relationships.

At the heart of LLMs lies the concept of tokens, which serve as the elemental units of text. Tokens can be individual words, subword units, or characters. Let $x_{1:n} := \{x_1, x_2, \ldots, x_n\}$ represent a sequence of

tokens. The goal of an LLM is to estimate the probability distribution of the next token x_{n+1} conditioned on the previous tokens $x_{1:n}$

$$\mathbb{P}\left[x_{n+1} \mid \{x_1, x_2, \ldots, x_n\}\right].$$

An LLM is a neural network architecture designed to learn and approximate this conditional probability distribution over sequences of tokens with a large number of parameters Θ. Namely, we can formulate an LLM as a parameterized function f_Θ that maps a sequence of tokens $\{x_1, x_2, \ldots, x_n\}$ to a probability distribution over the vocabulary, where the parameters Θ are learned from a large corpus of text training data.

$$f_\Theta : \{x_1, x_2, \ldots, x_n\} \to \mathbb{P}\left[x_{n+1} \mid \{x_1, x_2, \ldots, x_n\}; \Theta\right]$$

Interacting with an LLM involves specifying a prefix sequence $x_{1:n}$, termed the "*prompt*", and sampling the subsequent tokens $x_{n+1:z}$, known as the "*completion*". This process enables users to guide and control the generation of text according to desired contexts and constraints.

$$\underbrace{\{x_1, \ldots, x_n\}}_{\text{prompt}} \longrightarrow \underbrace{\{x_{n+1}, \ldots, x_z\}}_{\text{completion}}$$

4.2.2 Evolution of LLMs

The transformer architecture, introduced in the seminal work "*Attention Is All You Need*" ([30]), revolutionized LLM development due to its superior handling of long-range dependencies and efficient parallelization of computations. Subsequent advancements include the encoder-only `BERT` model ([31]), showcasing the power of pre-training on large datasets for fine-tuning on specific tasks.

Conversely, OpenAI's `GPT` series ([32]) demonstrated the potential of decoder-only models for generative tasks. In particular, the release of `GPT-3` marked a significant leap in LLM capabilities with its 175 billion parameters and remarkable few-shot learning abilities. This model highlighted the importance of prompt engineering, where carefully crafted prompts can guide model outputs without extensive fine-tuning.

The trend towards open-source models like `BLOOM` ([33]), `Mixtral` and Meta's `Llama` series ([34]) emphasizes accessibility and transparency in LLM development. The latest models, including OpenAI's `GPT-4.5`, `GPT-o1` and `o3`, Google's `Gemini 2.5 Pro`, Anthropic's `Claude 3.7 Sonnet`, and Meta's `Llama-3` series continue to push boundaries with improved accuracy, multimodal capabilities, and larger context windows.

4.2.3 Function Calling with Llama-3

In our endeavor we will employ `Llama-3`, developed by Meta AI and released on April 18, 2024[1] . This model has been pre-trained on approximately 15 trillion tokens of text gathered from "publicly available

[1] "*Introducing Meta Llama 3: The most capable openly available LLM to date*" [April 18, 2024]

sources" and it comes in two sizes: 8 billion and 70 billion parameters. In this application, we will employ the 70B version, which we will access through an API via `GroqCloud`.

Moreover, we will employ a *function calling* approach to streamline the process of interacting with the LLM. This implies prespecifying a set of functions to the LLM that will then be passed through our dataset of news articles to obtain a structured output in `JSON` format. The formal procedure is thoroughly described in Appendix Algorithm 4.

Each article $i \in \mathcal{D}$ implies a conversation with the LLM. The structure of the conversation consists of defining first a "system message", which provides a general context and purpose to the model. In our case:

– *You are a function calling LLM that analyses business news in Spanish.*
– *For every article, you must identify the firms directly affected by the news. Do not include every firm mentioned in the article, only include those that are directly affected by the shocks narrated therein.*
– *The identified firms must be Spanish and should be publicly listed in the Spanish exchange (their ticker is of the form 'TICKER.MC'). Do not include non-Spanish foreign firms. Do not include Spanish firms that are not publicly traded.*
– *For each identified firm, classify the shocks that affect them (type, magnitude, category). The type of shock can be 'demand', 'supply', 'financial', 'policy', or 'technology'. The magnitude can be 'minor' or 'major'. The direction can be 'positive' or 'negative'.*
– *If a firm is affected neutrally by the news article, don't include it in the analysis.*

Then, a news article is fed to the LLM. For illustration purposes, we will work with Example 2:

Example 2: An article about Cellnex and Telefónica (translated into English)

Cellnex will face more competition in Europe

Telefónica's (TEF.MC) subsidiary, Telxius Telecom, has agreed to sell its telecommunications tower division in Europe and Latin America to American Tower (AMT), which will expand the latter's presence in Europe and increase competition for the Spanish wireless telecommunications group Cellnex Telecom (CLNX.MC), according to Equita Sim. The transaction "represents the entry of a new independent tower operator into the Spanish market and potentially more competition for future growth in the European market as well," says the brokerage firm.

Next, we define an umbrella function "*firms*", which asks the LLM to identify the set $\mathcal{F}^i_{LLM}$ for each $i \in \mathcal{D}$. Then, for each $j \in \mathcal{F}^i_{LLM}$ we ask the LLM to categorize the type, expected magnitude, and expected direction that the shock described in the article implies in that particular firm j.

[Insert Table 2 about here]

The function calling schema is outlined in Table 2. First, we need to prompt the LLM, and then we need to specify the desired format of its response. The "Options" column imposes the answer format that the LLM must follow. For example, in `firms`, the "`array`" option indicates that the answer must be an enumeration of firms, while the "`string`" option in the subfunctions `firm` and `ticker` indicates that the answer must be a single name. Finally, the `shock_` subfunctions ask the LLM to choose from a predefined set of possible responses.

Note that the firms identified by the LLM are used to validate the firms identified by the pattern recognition algorithm (those extracted with `regex` by exploiting the pattern `<WORD>.MC`). As mentioned earlier, given the high quality of the filtered dataset (the ticker of the firms that are actively involved in the article are explicitly stated), they are almost identical. Hence, we indistinctively use $\mathcal{F}_i$ to simplify notation.

The LLM provides two outputs: structured data ("*Structured Output*") and a explanatory text describing its reasoning ("*Unstructured Ouptut*"). The explanations help us verify if the model correctly understands how to use the function-calling schema and follow system instructions. To assess the LLM's understanding, we review a random sample of these explanations and look for patterns of misinterpretation, confusion, or hallucination. If we identify such issues, we refine the system prompts and function descriptions to provide clearer guidance. This iterative prompt refinement continues until the LLM reliably generates correct outputs across multiple test scenarios.

1) Structured Output:

`firm`	`ticker`	`shock_type`	`shock_magnitude`	`shock_direction`
Cellnex Telecom	CLNX.MC	supply	minor	negative
Telefónica	TEF.MC	financial	minor	positive

2) Unstructured Output (justification)

The news about American Tower's expansion in Europe may increase competition for Cellnex, which is why it's classified as a negative supply shock. On the other hand, Telefónica benefits from the sale of its tower division, which is why it's classified as a positive financial shock.

This procedure is run iteratively from beginning (defining system prompt) to end (getting the output) for every $i \in \mathcal{D}$.[2]

[2] This procedure was run on a MacBook Pro M2 with 16GB RAM, 12-core central processing units (CPU), 19-core graphics processing units (GPU), and 16-core Neural Engine.

4.2.4 Clustering with the LLM

Formally, we can define the set $\mathcal{B} := \{(i, j) \mid i \in \mathcal{D} \wedge j \in \mathcal{F}^i\}$ containing all the unique pairs of articles and identified firms. The LLM assigns each pair $(i, j) \in \mathcal{B}$ with a choice from each of the following sets:

$$\begin{aligned} &\text{“}\textit{shock type}\text{”} && \mathcal{S}_T && := \{\text{demand, supply, financial, technology, policy}\} \\ &\text{“}\textit{shock magnitude}\text{”} && \mathcal{S}_M && := \{\text{minor, major}\} \\ &\text{“}\textit{shock direction}\text{”} && \mathcal{S}_D && := \{\text{positive, negative}\} \end{aligned}$$

The clustering of news articles follows naturally by taking the Cartesian product of these three sets: $\mathcal{G}_{LLM} := \mathcal{S}_T \times \mathcal{S}_M \times \mathcal{S}_D$, and the total number of clusters is now $k_{LLM} = |\mathcal{G}_{LLM}| = 20$. Consequently, a news article to which the LLM assigns $s_T \in \mathcal{S}_T$, $s_M \in \mathcal{S}_M$, $s_D \in \mathcal{S}_D$ will belong to cluster $(s_T, s_M, s_D) \in \mathcal{G}_{LLM}$. Formally, the set of all possible clusters is defined as:

$$\mathcal{G}_{LLM} := \{(s_T, s_M, s_D) \mid s_T \in \mathcal{S}_T, s_M \in \mathcal{S}_M, s_D \in \mathcal{S}_D\},$$

and each cluster can then be mapped to a positive integer as $\mathcal{G}_{LLM} \to \{k \in \mathbb{N}_0 \mid 0 \leq k \leq 19\}$. A representative sample of 3 articles from each cluster is provided in Appendix Table A2.

In Figure 7 we plot the distribution of news articles through clusters. As we can see, most articles are assigned to clusters 8, 9, 10, and 11, which are the clusters referred to financial events or shocks. Such clusters are mostly composed of articles about the publication of quarterly and semiannual results. More specifically, cluster 8 (*financial, minor, positive*) concentrates around 1/3 of the sample and is associated to the publication of results that mildly surpass the expectations of investors, hence, making this cluster a good candidate for a long trading signal.

On the other hand, other clusters such as 16 (*policy, minor, positive*) and 0 (*demand, minor, positive*) also concentrate a big share of news. Note that no cluster has been assigned to cluster 13 (*technology, minor, negative*). Compared to KMeans clustering with embeddings, the distribution of articles across these refined clusters is now remarkably stable across different data splits. This consistency indicates that clustering based on a thorough analysis of the shocks implied by each article for the affected firms yields a robust, time-invariant categorization. This is an encouraging finding for subsequent research and applications.

[Insert Figure 7 about here]

5. Trading Strategy

5.1 Beta-neutral positions on every $(i, j) \in \mathcal{B}$

Since we are interested in the individual effect of an article $i \in \mathcal{D}$ in each of the affected firms $j \in \mathcal{F}^i$, we work with the set $\mathcal{B} := \{(i, j) \mid i \in \mathcal{D} \wedge j \in \mathcal{F}^i\}$, where $|\mathcal{B}| = 3410 > |\mathcal{D}| = 2613$. We then fit a

market model to each unique pair $(i, j) \in \mathcal{B}$ on a lookback window of 100 days with a buffer of 10 days before the effective treatment date $\tilde{d}_0^i$.

$$r_d^j = \alpha^{(i,j)} + \beta^{(i,j)} r_d^M + \epsilon_d^{(i,j)},$$

where r_d^j denotes the return of firm j at trading day d in excess of the risk-free asset, which we take to be the daily euro short-term rate (€STR), and r_d^M denotes the excess return of the market (IBEX-35). These returns are obtained from adjusted close prices, which correct the price evolution for corporate actions such as dividends, stock splits, and new stock issuance. The notation overload in the regression coefficients $(\alpha^{(i,j)}, \beta^{(i,j)})$ emphasizes the fact that α and β are specific to each pair $(i, j) \in \mathcal{B}$ since the market model is computed for each firm $j \in \mathcal{F}_{\text{IBEX-35}}$ on a lookback window of time which is particular to each article $i \in \mathcal{D}$.

The reason why we fit a market model to each $(i, j) \in \mathcal{B}$ is to then apply a market-neutral strategy as in [26] and [35]. This is an investment approach designed to minimize or eliminate exposure to overall market movements, isolating the performance of a specific firm. In particular, we employ a beta-neutral strategy by buying one unit of firm j's stock and shorting $\beta^{(i,j)}$ units of the market index (i.e.: an ETF replicating the IBEX-35). This hedged position harvests the idiosyncratic returns from the market model and it only makes sense when firm j's returns are expected to outperform or underperform the market.[3] The position delivers abnormal returns $AR_d^{(i,j)}$ at some trading day $d \geq \tilde{d}_0^i$ given by

$$r_d^j - \beta^{(i,j)} r_d^M = \alpha^{(i,j)} + \epsilon_d^{(i,j)} =: AR_d^{(i,j)}.$$

The position is taken at the effective treatment date $\tilde{d}_0^i$ and is maintained over a holding window consisting of $L \in \mathbb{N}$ trading days after $\tilde{d}_0^i$, where L is set to 4 trading days. The justification for this choice of L results from the maximization of the Sharpe Ratio of the portfolio in the train and validation samples for both KMeans and LLM-based clustering.[4] Finally, we compute the Sharpe Ratio of each position $SR^{(i,j)}$, which we will subsequently employ to optimize cluster selection.

5.2 Optimal Cluster Selection

After taking beta-neutral positions on each pair $(i, j) \in \mathcal{B}$ and holding them over L days, we can obtain a measure of how profitable the positions are on average for articles that belong to the same cluster. For this purpose, let $\mathcal{B}_g$ denote the set of all article-firm pairs such that the article belongs to some cluster $g \in \mathcal{G}$.

$$\mathcal{B}_g := \{(i, j) \mid (i, j) \in \mathcal{B} \ \wedge \ i \in \mathcal{D}_g\}.$$

[3] For expected underperformance of firm j, reverse the beta-neutral positions: sell one unit of firm j and buy $\beta^{(i,j)}$ units of the market index. However, note that this will be handled later by a Trading Rule (TR).

[4] The choice of L is justified in detail in A.2, and the sensitivity of the trading strategy's out-of-sample performance to different values of L is examined in Section 6 ("*Robustness Checks*").

The average Sharpe Ratio associated to each cluster is

$$\overline{SR}_g = \frac{1}{|\mathcal{B}_g|} \sum_{(i,j)\in\mathcal{B}_g} SR^{(i,j)},$$

and it provides a measure of the performance of the beta-neutral positions in each cluster. The distribution of cluster-average Sharpe Ratios across the different clusters is shown in Appendix Figure A5.

We then focus on developing two algorithms that optimally leverage the cluster information for our trading strategy. Our approach draws parallels with traditional portfolio sorting methods, where assets are typically arranged into deciles based on specific characteristics, and trading positions are established by going long on top deciles and short on bottom ones. Similarly, our strategy will construct self-financing portfolios based on clusters rather than individual assets: taking long positions in clusters expected to outperform and short positions in those expected to underperform. To identify the optimal clusters for trading, we propose two distinct algorithmic approaches. The first approach, which we term "*greedy*", selects clusters by maximizing the Sharpe Ratio within the validation dataset. The second approach, termed "*stable*", utilizes a broader information set by incorporating both training and validation data, aiming to identify clusters that maintain consistent performance across both splits. In both algorithms, we impose sign restrictions to ensure that our trading positions align with the expected direction of returns.

5.2.1 Greedy Algorithm

The greedy selection of clusters is done in the validation sample $\mathcal{B}^{val} := \{(i,j) \in \mathcal{B} \mid i \in \mathcal{D}^{val}\}$, from where we compute the cluster-average $\overline{SR}_g^{val}$ for each $g \in \mathcal{G}$. Define $\mathcal{G}_{SR^+}^{val} := \{g \in \mathcal{G} \mid \overline{SR}_g^{val} > 0\}$ and $\mathcal{G}_{SR^-}^{val} := \{g \in \mathcal{G} \mid \overline{SR}_g^{val} < 0\}$ as the sets of clusters with positive and negative Sharpe Ratios in the validation sample. Obviously, we will be interested in taking long positions when reading an article that is clustered in some $g \in \mathcal{G}_{SR^+}^{val}$, and short positions in clusters $g \in \mathcal{G}_{SR^+}^{val}$. However, our trading strategy will not trade every cluster $g \in \mathcal{G}$. Instead, it will select the clusters from $\mathcal{G}_{SR^+}$ and $\mathcal{G}_{SR^-}$ that lead the to most profitable trades. To identify such clusters, we rank them by their average Sharpe Ratio. Define the ranking function $\mathfrak{R} : \mathcal{G} \to \{1, \ldots, k^*\}$ such that

$$\mathfrak{R}_g^{val} = \sum_{h\in\mathcal{G}} \mathbf{1}\left(\overline{SR}_h^{val} \geq \overline{SR}_g^{val}\right) ,$$

where $\mathbf{1}(\cdot)$ is the indicator function which equals 1 if the condition inside is true and 0 otherwise.

The number of traded clusters on either side (long and short) will be upper-bounded by some hyper-parameter of our choice $\theta \in \mathbb{N}$ which we set proportional to the number of clusters. Namely, $\theta = \lfloor \rho k \rfloor$ for some $\rho \in (0,1)$, which has been set to $\rho = 0.5$ to maximize the Sharpe Ratio of the trading strategy in the training and validation samples [5]. The actual number of traded clusters will not be exactly θ as

[5] The choice of θ is justified in detail in A.2. The sensitivity of the trading strategy's out-of-sample performance to different values of θ is examined in Section 6 ("*Robustness Checks*").

there is a natural bound coming from the cardinalities of $\mathcal{G}_{SR^+}$ and $\mathcal{G}_{SR^-}$. Hence, the actual number of long and short-traded clusters will be $\theta^+ := \min(\theta,\ |\mathcal{G}_{SR^+}|)$ and $\theta^- := \min(\theta,\ |\mathcal{G}_{SR^-}|)$. The set of traded clusters $\mathcal{G}_\theta$ is defined as

$$\mathcal{G}_\theta := \left\{ g \in \mathcal{G} \mid 1 \leq \mathfrak{R}_g^{val} \leq \theta^+ \ \vee \ k^* - \theta^- < \mathfrak{R}_g^{val} \leq k^* \right\} = \mathcal{G}_\theta^+ \cup \mathcal{G}_\theta^- \ ,$$

where $\mathcal{G}_\theta^+ := \{g \in \mathcal{G} \mid 1 \leq \mathfrak{R}_g^{val} \leq \theta^+\}$ is the set of long-traded clusters, $\mathcal{G}_\theta^- := \{g \in \mathcal{G} \mid k^* - \theta^- < \mathfrak{R}_g^{val} \leq k^*\}$ is the set of short-traded clusters and, clearly, $|\mathcal{G}_\theta| = \theta^+ + \theta^-$.[6] In Appendix Algorithm 2., we can find the formal design of this algorithm.

5.2.2 Stable Algorithm

In this case, we prioritize the stability of the cluster rankings by ensuring that the traded clusters minimize the rank difference of the cluster-average Sharpe Ratios between the training and validation samples. To begin, we compute the rank of each cluster based on the average Sharpe Ratios in both the training and validation samples. This delivers $\{\mathfrak{R}_g^{tr}\}_{g\in\mathcal{G}}$ and $\{\mathfrak{R}_g^{val}\}_{g\in\mathcal{G}}$, which provides a measure of the relative performance of the clusters within each sample.

Next, we calculate the absolute difference in ranks between the training and validation samples for each cluster, which allows us to measure the stability of each cluster's performance between the two samples

$$\delta_g := |\mathfrak{R}_g^{tr} - \mathfrak{R}_g^{val}| \ .$$

Clusters are then sorted based on their rank differences δ_g in descending order. To do this, we can simply compute the ranking of the ranking differences as

$$\mathfrak{R}(\delta_g) := \sum_{h\in\mathcal{G}} \mathbf{1}\left(\delta_g \geq \delta_h\right).$$

Next, we select the top $2\theta \in \mathbb{N}$ clusters with the smallest rank differences, indicating the most stable clusters across the training and validation samples. The selected clusters now are

$$\mathcal{G}_\theta = \{g \in \mathcal{G} \mid 1 \leq \mathfrak{R}(\delta_g) \leq 2\theta\}.$$

Finally, we determine the sets of long and short-traded clusters based on the average Sharpe Ratios in both the training and validation samples. In particular, the set of long-traded clusters ($\mathcal{G}_\theta^+$) are the ones that have positive average Sharpe Ratios in both, training and validation samples

$$\mathcal{G}_\theta^+ = \{g \in \mathcal{G}_\theta \mid \overline{SR}_g^{tr} > 0 \ \wedge \ \overline{SR}_g^{val} > 0\},$$

[6] Alternatively, we could trade the same number of clusters in the long and short side by defining a unique $\theta^* := \min\left(\theta, |\mathcal{G}_{SR^+}|, |\mathcal{G}_{SR^-}|\right)$ such that $\mathcal{G}_\theta := \left\{ g \in \mathcal{G} \mid 1 \leq \mathfrak{R}_g^{val} \leq \theta^* \ \vee \ k^* - \theta^* < \mathfrak{R}_g^{val} \leq k^* \right\}$ and $|\mathcal{G}_\theta| = 2\theta^*$.

and by symmetry, short-traded clusters ($\mathcal{G}_\theta^-$) are the ones that have negative average Sharpe Ratios in both, training and validation samples

$$\mathcal{G}_\theta^- = \{g \in \mathcal{G}_\theta \mid \overline{SR}_g^{tr} < 0 \ \wedge \ \overline{SR}_g^{val} < 0\} \ .$$

This approach ensures that we select the most stable clusters for trading, reducing the risk associated with rank variability between the training and validation samples, and ensuring that the direction of the signal is consistent across the two splits. The final output consists of the sets of long-traded and short-traded clusters, which are then used to implement the trading strategy. The implementation of the algorithm is methodically presented in Appendix Algorithm 3.

[Insert Table 3 about here]

In Table 3 we show the 26 clusters with their proposed names (based on the articles they pool together as shown in Appendix Table A1) and the selection of long and short-traded clusters according to each algorithm: "*greedy*" and "*stable*". We write "LONG" for those clusters $g \in \mathcal{G}_\theta^+$ and "SHORT" for $g \in \mathcal{G}_\theta^-$. As we can see, trading clusters of news articles based on this procedure is quite risky, as there is a high reliance of the signal on the past performance of a cluster. For example, clusters 21 and 22 are linked to the financial performance of Repsol and Aena, respectively, during the training and validation samples. Evidently, the future performance of these firms can change, but the signal provided by the algorithm will still indicate "LONG". Additionally, some clusters are heavily built on specific events of the period of time they were constructed upon. For example, cluster 17 pools articles related to the challenges of the tourism industry in Spain in Covid times, and cluster 25 is related to the post-covid developments of Inditex and Acerinox. Thus, a clustering approach based on embeddings is not generalizable over time. As the world evolves, clusters become outdated and require constant recalibration to maintain their relevance and predictive power. Hence, any trading strategy based solely on historical cluster performance is likely to produce misguided trading signals over time

[Insert Table 4 about here]

In contrast, our LLM-based clustering methodology offers significant advantages by focusing on the fundamental nature of economic shocks rather than historical patterns. This approach provides more robust and generalizable signals that are less susceptible to temporal changes in market conditions. Moreover, unlike the black box nature of vector embeddings, our methodology offers transparency and interpretability in signal generation. This is evident in how the *Greedy* algorithm's cluster selection closely aligns with the direction of economic shocks: negative shocks typically correspond to price decreases and positive shocks to increases.

Looking at Table 4, we observe that both algorithms consistently short articles classified as policy shocks, regardless of direction, while going long on cluster 8, which contains approximately one-third

of news articles (those categorized as undergoing financial minor and positive shocks). This consistent shorting of policy shocks likely reflects markets' general aversion to policy uncertainty, as policy changes –even positive ones– often create implementation uncertainty and take time for market participants to fully price in. Interestingly, both algorithms also exhibit seemingly counter-intuitive behavior by going long on negative major demand shocks and short on positive major demand shocks. This pattern might suggest a "mean reversion" expectation in the algorithms, where major demand shocks are viewed as temporary deviations that will eventually correct: negative shocks present buying opportunities, while positive shocks signal potential overvaluation.

5.3 Trading Rule & Portfolio Construction

For a given selection of clusters $\mathcal{G}_\theta^+$ and $\mathcal{G}_\theta^-$, we launch trades and hold them for $L = 4$ trading days. Formally, the trading rule for a pair $(i,j) \in \mathcal{B}$ at trading day $d \in \tilde{\mathfrak{d}}$ is

$$TR_{L,\theta}\langle(i,j),d\rangle := \begin{cases} +1 & \text{if} \quad [(i,j) \in \mathcal{B}_g \ \wedge \ g \in \mathcal{G}_\theta^+] \ \wedge \ d \in (\tilde{d}_0^i, \tilde{d}_0^i + L] \\ 0 & \text{if} \quad [(i,j) \in \mathcal{B}_g \ \wedge \ g \notin \mathcal{G}_\theta \ ] \ \vee \ d \notin (\tilde{d}_0^i, \tilde{d}_0^i + L] \\ -1 & \text{if} \quad [(i,j) \in \mathcal{B}_g \ \wedge \ g \in \mathcal{G}_\theta^-] \ \wedge \ d \in (\tilde{d}_0^i, \tilde{d}_0^i + L] \end{cases} .$$

In this context, a portfolio is a collection of positions taken in a firm's stocks according to $TR_{L,\theta}\langle(i,j),d\rangle$. In other words, it is the set of all $\langle(i,j),d\rangle$ for which a trade is executed.

$$\mathcal{P} := \left\{ \langle(i,j),d\rangle \mid (i,j) \in \mathcal{B} \ \wedge \ d \in \tilde{\mathfrak{d}} \ \wedge \ TR_{L,\theta}\langle(i,j),d\rangle \neq 0 \right\} .$$

The set of open positions on a particular day $d \in \tilde{\mathfrak{d}}$ is defined as

$$\mathcal{P}_d := \{(i,j) \in \mathcal{B} \mid TR_{L,\theta}\langle(i,j),d\rangle \neq 0\} ,$$

and the portfolio is rebalanced every day, so each position $(i,j) \in \mathcal{P}_d$ receives a weight that is inversely proportional to the total amount of open positions in that day (i.e. $1/|\mathcal{P}_d|$).[7] This produces an equally-weighted rolling-portfolio similar to [36] and [26]. The overlapping returns of the portfolio at $d \in \tilde{\mathfrak{d}}$ can be obtained as an average of the abnormal returns weighted by the trading rule, which determines the direction of each position (long or short), and scaled by the number of open positions in that day,

$$r_d^{\mathcal{P}} := \frac{1}{|\mathcal{P}_d|} \sum_{(i,j),\in \mathcal{P}_d} TR_{L,\theta}\langle(i,j),d\rangle \cdot AR_d^{(i,j)} .$$

In Figure 8 we plot the cumulative gross returns of trading strategies based on KMeans clustering (Panel A) and LLM clustering (Panel B) across different data splits

[7] Note that the cardinality of the set of open positions at day $d \in \tilde{\mathfrak{d}}$, denoted as $|\mathcal{P}_d|$, can be computed as the sum of the absolute values of the trading rule over all pairs $(i,j) \in \mathcal{B}$ for a given trading day $d \in \tilde{\mathfrak{d}}$.

$$|\mathcal{P}_d| = \sum_{(i,j)\in\mathcal{B}} |TR_{L,\theta}\langle(i,j),d\rangle| .$$

[Insert Figure 8 about here]

KMeans. In panel A of Table 5 we show the portfolio statistics of the benchmark model. As we can see, both algorithms work well on the data splits they were trained on: the Stable algorithm works well on both, training and validation data, while the Greedy algorithm does a good job only on validation data as expected. However, this doesn't say anything about any of these algorithms, as it is easy to make profitable trades *in-sample*. The generalizability of the strategy is determined *out-of-sample* in the test data. The empirical analysis reveals significant challenges in the strategy's ability to maintain consistent performance across different time periods. During the training and validation phases, the methodology shows promising results with annualized returns ranging from 26.6% to 47.7% and strong risk-adjusted performance metrics (Sharpe ratios between 2.0 and 3.2). However, this performance deteriorates substantially in the test period, where returns drop to modest levels (2.9% to 4.9% annually) with significantly lower Sharpe ratios (0.2 to 0.7), suggesting that the strategy's alpha-generating capability does not generalize well out of sample. The distributional properties of returns in the test period provide additional insights into the strategy's behavior under true out-of-sample conditions. The shift from negative to strongly positive skewness (1.85 to 2.46) coupled with high excess kurtosis (5.50 to 14.57) suggests that the strategy's return distribution has fundamentally changed, characterized by more frequent small losses offset by occasional large gains. This asymmetric return pattern, while potentially appealing from a risk preference perspective, differs markedly from the training period characteristics. The tail risk measures further illuminate the strategy's risk profile, with annualized 95% VaR ranging from -7.8% to -18.9% and corresponding CVaR from -9.7% to -26.8% in the test period. These statistical properties, combined with the strong dependence on historical cluster-specific performance, indicate that the strategy fails to identify stable and generalizable trading signals, likely due to its reliance on firm and industry-specific clustering patterns that do not persist out of sample. As we can see in the plot, neither algorithm is able to generate a consistent profile of earnings, and the statistics confirm that profits are negligible, and would likely be eaten away by exogenous market frictions (e.g. trading costs).

[Insert Table 5 about here]

LLM. Panel B of Table 5 presents the performance metrics for our LLM-based approach. As before, both algorithms perform really well on "seen" data. However, different from before, the *Greedy* algorithm works well also on the Training Split (which it was not trained on). More importantly, both algorithms do a great job in the test data. As we can see, both are able to achieve a consistent profile of earnings through the split. The portfolio statistics reveal notable consistency in the strategy's performance across different time periods. During the training and validation phases, the methodology demonstrates solid performance with annualized returns ranging from 16.0% to 28.3% and Sharpe ratios between 1.4 and 2.9. This performance strengthens in the test period, where returns increase to 30.8%-37.2% annually with Sharpe ratios of 4.3-4.4, indicating that the strategy's alpha-generating capability successfully generalizes

to out-of-sample conditions. The distributional properties of returns provide evidence for the strategy's robustness. The test period maintains positive skewness (0.84 to 1.49) and moderate to high excess kurtosis (1.95 to 8.30), indicating an asymmetric return pattern with more frequent small losses offset by larger gains. This return distribution is complemented by contained maximum drawdowns (1.1% to 1.5%) and strong Calmar ratios (21.0 to 34.5) in the test period. The tail risk measures further support the strategy's risk management properties, with annualized 95% VaR ranging from -6.9% to -9.5%, and CVaR ranging from -9.9% to -11.3% in the test period. Taken together, the strategy's ability to sustain consistent out-of-sample performance metrics demonstrates that the LLM-based clustering approach identifies enduring trading signals that transcend specific market regimes.

While our primary focus has been on developing a methodology to anticipate market reactions to news (i.e., identifying winners and losers to assess the predictive power of our LLM-based approach), we also analyze the trading intensity and implementation costs of the resulting strategies. The detailed examination in A.8 reveals that, after accounting for transaction costs, the LLM-based approach maintains its superior performance relative to KMeans, though with attenuated profitability. Therefore, practitioners interested in the practical implementation of this strategy would benefit from optimizing the trading strategy by incorporating transaction costs into their framework.

6. Robustness Checks

In our applications we have worked with a holding period of $L = 4$ trading days and an upper bound on traded clusters of $\theta = \lfloor 0.5k \rfloor$. As shown in A.2, such choices result from the maximization of the Sharpe Ratios in the train and validation samples. All that is left is to check whether our out-of-sample results are sensitive to the choice of hyperparameters (L, θ). For this purpose, we evaluate the variability of the Sharpe Ratios of the test portfolio ($SR^{\mathcal{P}^{test}}$) to changes in L and θ.

First, we focus on the holding period length of the beta-neutral strategy (L). For this purpose, we fix $\theta = \lfloor 0.5k \rfloor$ and, for each clustering method, obtain the series of Sharpe Ratios over a grid $\mathbf{L}$ (which ranges from 1 to 20 trading periods). This delivers the series $\{SR^{\mathcal{P}^{test}}(L)\}_{L \in \mathbf{L}}$, which we then plot in two formats. On the left side of Figure 9 we plot the distribution of Sharpe Ratios in the grid, and in the right side, we show the mapping $L \mapsto SR^{\mathcal{P}^{test}}(L)$ over $\mathbf{L}$.

[Insert Figure 9 about here]

From Figure 9a it follows that KMeans clustering produces a distribution that is clearly left-skewed, while the distribution of $SR^{\mathcal{P}^{test}}$ for LLM clustering is clearly right-skewed (Figure 9c). This confirms the fact that LLM clustering generates Sharpe Ratios that are statistically higher than those generated by KMeans. The plots in the right-hand-side substantiate this observation: KMeans is only able to produce positive $SR^{\mathcal{P}^{test}}$ for really short holding window lengths (Figure 9b), while LLM clustering, although

not always stable, is, in general, able to produce positive Sharpe Ratios more consistently over the grid (Figure 9d).

We then turn to analyze the sensitivity of $SR^{\mathcal{P}^{test}}$ to different values for the upper bound on the number of traded clusters (θ). Now we fix $L = 4$ and define a grid $\boldsymbol{\theta}$, from where we can obtain $\{SR^{\mathcal{P}^{test}}(\theta)\}_{\theta \in \boldsymbol{\theta}}$.

[Insert Figure 10 about here]

The results of this exercise are shown in Figure 10. As we can see, in Figure 10a the results are mixed for the case of KMeans clustering. Namely, the *Stable* algorithm is able to generate positive Sharpe Ratios but the *Greedy* algorithm struggles to do so. In Figure 10b we see what is happening: *Stable* works well with for low values of θ, while *Greedy* only works for high values of θ. This high reliance of the algorithms on specific values of θ points to the instability of the trading strategy when employing KMeans clustering.

On the other hand, Figure 10c shows a clear pattern for the case of LLM clustering. Namely, the mass accumulates at high and positive Sharpe Ratios. This observation is further substantiated by Figure 10b, which shows that leaving aside the fact that the greedy algorithm does bad for really low values of θ (i.e.: $\theta \leq 3$), in general, the trading strategy is now able to produce high, positive and stable Sharpe Ratios across different values of θ.

All in all, our results are robust to hyperparameter variability, showing that LLM clustering consistently beats a strategy based on clustering embeddings with KMeans.

7. Conclusion

This paper investigates how information from business news affects stock market prices. We analyze a dataset of Spanish business articles during a particularly volatile period-the COVID-19 pandemic-and examine firm-specific stock market reactions to news. We show that transforming text into vector embeddings and clustering them using KMeans yields clusters that are firm-specific and industry-specific. However, the distribution of articles across clusters is unstable over sequential data splits, indicating temporal instability. When we implement a cluster-based trading strategy-similar to portfolio sorts-on the KMeans clusters, we observe an over-reliance on the past performance of a cluster. That is, signals are short-lived due to temporal instability. Consequently, the out-of-sample profitability of the trading strategy is negligible, evidencing the method's poor temporal generalizability. Therefore, a model based on embeddings is superficial and is not able to anticipate market trends.

As an alternative, we develop a novel approach by guiding a Large Language Model (LLM) through a structured news-parsing schema, enabling it to analyze news-implied firm-specific economic shocks. The schema involves identifying the firms affected by the articles and classifying the implied shocks on such firms by their type, magnitude, and direction. This LLM-based methodology demonstrates several advantages over the traditional clustering approach. Even in a volatile period, it produces stable distributions

of articles across clusters in sequential splits, demonstrating robust temporal stability. Moreover, the resulting trading signals are both long-lasting and economically relevant, as they are based on fundamental economic shocks rather than statistical patterns. The results show that the LLM-based trading strategy effectively identifies winners and losers, illustrating the parser's ability to anticipate market trends by comprehending the economic implications of firm-specific shocks. This approach generates a consistent profile of earnings in the test set, with results robust to the choice of hyperparameters-the holding period length of the trading strategy and the number of selected clusters for trading. Our findings demonstrate a promising avenue: LLMs, when guided by appropriate economic frameworks, can help predict market reactions to news through systematic classification of economic shocks embedded in financial narratives.

References

[1] E. F. Fama, Efficient capital markets: A review of theory and empirical work, J. Finance 25 (2) (1970) 383. doi:10.2307/2325486.
URL http://dx.doi.org/10.2307/2325486

[2] P. C. Tetlock, Giving content to investor sentiment: The role of media in the stock market, J. Finance 62 (3) (2007) 1139–1168. doi:10.1111/j.1540-6261.2007.01232.x.
URL http://dx.doi.org/10.1111/j.1540-6261.2007.01232.x

[3] P. C. Tetlock, M. Saar-Tsechansky, S. Macskassy, More than words: Quantifying language to measure firms' fundamentals, J. Finance 63 (3) (2008) 1437–1467. doi:10.1111/j.1540-6261.2008.01362.x.
URL http://dx.doi.org/10.1111/j.1540-6261.2008.01362.x

[4] T. Loughran, B. McDonald, When is a liability not a liability? textual analysis, dictionaries, and 10ks, J. Finance 66 (1) (2011) 35–65. doi:10.1111/j.1540-6261.2010.01625.x.
URL http://dx.doi.org/10.1111/j.1540-6261.2010.01625.x

[5] N. Jegadeesh, D. Wu, Word power: A new approach for content analysis, J. Financ. Econ. 110 (3) (2013) 712–729. doi:10.1016/j.jfineco.2013.08.018.
URL http://dx.doi.org/10.1016/j.jfineco.2013.08.018

[6] J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market, J. Comput. Sci. 2 (1) (2011) 1–Ş8. doi:10.1016/j.jocs.2010.12.007.
URL http://dx.doi.org/10.1016/j.jocs.2010.12.007

[7] D. Garcia, Sentiment during recessions, J. Finance 68 (3) (2013) 1267–1300. doi:10.1111/jofi.12027.
URL http://dx.doi.org/10.1111/jofi.12027

[8] Z. T. Ke, B. Kelly, D. Xiu, Predicting returns with text data, Tech. rep., National Bureau of Economic Research (Aug. 2019). doi:10.3386/w26186.
URL http://dx.doi.org/10.3386/w26186

[9] C.-C. Lee, Z. Gao, C.-L. Tsai, Bert-based stock market sentiment analysis, in: 2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), IEEE, 2020, pp. 1–Ş2. doi:10.1109/icce-taiwan49838.2020.9258102.
URL http://dx.doi.org/10.1109/icce-taiwan49838.2020.9258102

[10] F. Wei, U. Nguyen, Stock trend prediction using financial market news and bert, in: Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, SCITEPRESS - Science and Technology Publications, 2020, pp. 319–326. doi:10.5220/0010172103190326.
URL http://dx.doi.org/10.5220/0010172103190326

[11] W. Antweiler, M. Z. Frank, Do us stock markets typically overreact to corporate news stories?, SSRN Electron. J.doi:10.2139/ssrn.878091.
URL http://dx.doi.org/10.2139/ssrn.878091

[12] S. Hansen, M. McMahon, A. Prat, Transparency and deliberation within the fomc: A computational linguistics approach, Q. J. Econ. 133 (2) (2018) 801–Ş870. doi:10.1093/qje/qjx045.
URL http://dx.doi.org/10.1093/qje/qjx045

[13] L. Bybee, B. Kelly, A. Manela, D. Xiu, Business news and business cycles, J. Finance 79 (5) (2024) 3105–3147. doi:10.1111/jofi.13377.
URL http://dx.doi.org/10.1111/jofi.13377

[14] L. Bybee, B. Kelly, Y. Su, Narrative asset pricing: Interpretable systematic risk factors from news text, The Rev. Financ. Stud. 36 (12) (2023) 4759–4787. doi:10.1093/rfs/hhad042.
URL http://dx.doi.org/10.1093/rfs/hhad042

[15] G. Hoberg, G. Phillips, Text-based network industries and endogenous product differentiation, J. Polit. Econ. 124 (5) (2016) 1423–1465. doi:10.1086/688176.
URL http://dx.doi.org/10.1086/688176

[16] Q. Chen, Stock movement prediction with financial news using contextualized embedding from bert, arXiv preprint arXiv:2107.08721doi:10.48550/ARXIV.2107.08721.
URL https://arxiv.org/abs/2107.08721

[17] E. Benincasa, J. Fu, M. Mishra, A. Paranjape, Different shades of green: Estimating the green bond premium using natural language processing, SSRN Electron. J.doi:10.2139/ssrn.4198065.
URL http://dx.doi.org/10.2139/ssrn.4198065

[18] M. Jha, H. Liu, A. Manela, Does finance benefit society? a language embedding approach, Rev. Financ. Stud., Forthcomingdoi:10.2139/ssrn.3655263.
URL http://dx.doi.org/10.2139/ssrn.3655263

[19] C. L. Zhang, Feel the market: An attempt to identify additional factor in the capital asset pricing model (capm) using generative pre-trained transformer (gpt) and bidirectional encoder representations from transformers (bert), SSRN Electron. J.doi:10.2139/ssrn.4521946.
URL http://dx.doi.org/10.2139/ssrn.4521946

[20] X. Gabaix, R. S. J. Koijen, M. Yogo, Asset embeddings, SSRN Electron. J.doi:10.2139/ssrn.4507511.
URL http://dx.doi.org/10.2139/ssrn.4507511

[21] D. Cutler, J. Poterba, L. Summers, What Moves Stock Prices?, 1988. doi:10.3386/w2538.
URL http://dx.doi.org/10.3386/w2538

[22] M. L. Mitchell, J. H. Mulherin, The impact of public information on the stock market, J. Finance 49 (3) (1994) 923–950. doi:10.2307/2329211.
URL http://dx.doi.org/10.2307/2329211

[23] S. R. Baker, N. Bloom, S. J. Davis, Measuring economic policy uncertainty, Q. J. Econ. 131 (4) (2016) 1593–1636. doi:10.1093/qje/qjw024.
URL http://dx.doi.org/10.1093/qje/qjw024

[24] S. Baker, N. Bloom, S. Davis, M. Sammon, What Triggers Stock Market Jumps?, 2021. doi:10.3386/w28687.
URL http://dx.doi.org/10.3386/w28687

[25] A. Manela, A. Moreira, News implied volatility and disaster concerns, J. Financ. Econ. 123 (1) (2017) 137–162. doi:10.1016/j.jfineco.2016.01.032.
URL http://dx.doi.org/10.1016/j.jfineco.2016.01.032

[26] W. S. Chan, Stock price reaction to news and no-news: drift and reversal after headlines, J. Financ. Econ. 70 (2) (2003) 223–260. doi:10.1016/s0304-405x(03)00146-6.
URL http://dx.doi.org/10.1016/S0304-405X(03)00146-6

[27] P. Oncharoen, P. Vateekul, Deep learning for stock market prediction using event embedding and technical indicators, in: 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), IEEE, 2018, pp. 19–Ş24. doi:10.1109/icaicta.2018.8541310.
URL http://dx.doi.org/10.1109/icaicta.2018.8541310

[28] A. Lopez-Lira, Y. Tang, Can chatgpt forecast stock price movements? return predictability and large language models, SSRN Electron. J.doi:10.2139/ssrn.4412788.
URL http://dx.doi.org/10.2139/ssrn.4412788

[29] Y. Chen, B. T. Kelly, D. Xiu, Expected returns and large language models, Available at SSRN 4416687.
URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4416687

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30. doi:10.48550/ARXIV.1706.03762.
URL https://doi.org/10.48550/arxiv.1706.03762

[31] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805doi:10.48550/ARXIV.1810.04805.
URL https://arxiv.org/abs/1810.04805

[32] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training.

[33] T. Le Scao, A. Fan, C. Akiki, E. Pavlick, S. Ili, D. Hesslow, R. Castagn, A. S. Luccioni, F. Yvon, M. Gall, et al., Bloom: A 176b-parameter open-access multilingual language modeldoi:10.48550/ARXIV.2211.05100.
URL https://arxiv.org/abs/2211.05100

[34] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Roziére, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971.

[35] H. Jiang, S. Z. Li, H. Wang, Pervasive underreaction: Evidence from high-frequency data, J. Financ. Econ. 141 (2) (2021) 573–599. doi:10.1016/j.jfineco.2021.04.003.
URL http://dx.doi.org/10.1016/j.jfineco.2021.04.003

[36] N. Jegadeesh, S. Titman, Returns to buying winners and selling losers: Implications for stock market efficiency, J. Finance 48 (1) (1993) 65–91. doi:10.1111/j.1540-6261.1993.tb04702.x.
URL http://dx.doi.org/10.1111/j.1540-6261.1993.tb04702.x

TABLE 1: Summary Statistics of Articles by Data Split

Data Split	**Time Period**	**# Articles**	**# Words**	**Vocabulary Size**
Train	24/06/2020 – 12/02/2021	1254	327413	26762
Validation	12/02/2021 – 21/06/2021	836	232912	22265
Test	21/06/2021 – 30/09/2021	523	140495	16474
All	24/06/2020 – 30/09/2021	2613	700820	42603

Note: Summary statistics by data splits and for the whole sample. We provide the period spanned by each data split, the number of articles, the number of words, and the vocabulary size. Articles have been preprocessed following standard NLP practices.

TABLE 2: Function calling schema

Function	Prompt	Options
1. `firms`	"*List all the firms affected by the events narrated in the article*"	`array`
1.1. `firm`	"*Iterate over each* `firm` *in* `firms`"	`string`
1.2. `ticker`	"*State the stock market ticker of* `firm` "	`string`
1.3. `shock_type`	"*What type of shock does this article imply on* `firm` *?*"	{demand, supply, financial, technology, policy}
1.4. `shock_magnitude`	"*How much impact is this shock expected to have on* `firm`*?*"	{minor, major}
1.5. `shock_direction`	"*In what direction is this shock expected to impact* `firm`*?*"	{positive, negative}

This table outlines the structure of the function calling schema we designed to guide the LLM through the analysis of news-implied firm-specific economic shocks. The "Function" column specifices the name of the tool passed to the LLM. We can understand the umbrella function `firms` *as running a loop over each of its arguments, with the indented subfunctions being referred to the specific argument passed to them. The "Prompt" column provides an example of the simplified instructions given to the LLM (the actual prompts are longer as the LLM needs clear and detailed instructions, with useful examples for context). Finally, the "Options" column imposes the answer format that the LLM must follow. For example, in* `firms`*, the "*`array`*" option indicates that the answer must be an enumeration of firms, while the "*`string`*" option in the subfunctions* `firm` *and* `ticker` *indicates that the answer must be a single string. Finally, the* `shock_` *subfunctions ask the LLM to choose from a predefined set of possible responses.*

TABLE 3: Mapping of embeddings-based KMeans clusters to Trading Signals

	Cluster	Greedy	Stable
0	Miscellaneous (Colonial, Acciona, Amadeus, Grifols, Endesa, IAG, Bankinter...)	SHORT	
1	Quarterly & Semi-Annual Earnings Reports	SHORT	
2	BBVA & Sabadell: Financial Performance & Strategic Movements	SHORT	
3	Telefónica & Cellnex: Telecommunications Tower Sales & Market Dynamics	LONG	LONG
4	CaixaBank: Mergers and Strategic Moves in the Banking Sector		
5	Telefónica, Indra, & MásMóvil: Regulatory and Strategic Moves in Telecom	LONG	
6	Siemens Gamesa: Supply Agreements, Profitability Targets in Renewable Energy	SHORT	
7	Cellnex: Strategic Acquisitions and Financial Moves in Telecom Infrastructure	LONG	
8	Acciona, Endesa, Enagás & Naturgy: Strategic Moves & Regulatory Developments in the Energy Sector	LONG	
9	Repsol: Strategic Moves and Challenges in the Energy Sector	LONG	
10	Ferrovial, Acciona: Strategic Expansions and Financial Maneuvers in Infrastructure	SHORT	SHORT
11	Solaria: Strategic Moves and Market Challenges in Renewable Energy	LONG	LONG
12	Iberdrola: Strategic Collaborations and Renewable Energy Developments	SHORT	
13	IAG: Financial Performance	LONG	
14	Santander & CaixaBank: Financial Moves and Sustainability Initiatives	SHORT	
15	ACS & Acciona: Strategic Movements and Infrastructure Projects	SHORT	SHORT
16	Telefónica: Financial Performance and Strategic Moves	LONG	
17	Meliá and Spanish Tourism Sector: Challenges Amidst the Pandemic	SHORT	
18	Takeover Bids for Naturgy and MásMóvil	SHORT	
19	Naturgy: Financial Performance	SHORT	SHORT
20	PharmaMar, Grifols: Regulatory Approvals and Market Moves in the Pharmaceutical Sector	LONG	LONG
21	Repsol: Financial Performance	LONG	LONG
22	Aena: Financial Performance	LONG	LONG
23	Enagás, Endesa, Iberdrola, Red Eléctrica: Regulatory and Market Challenges in the Energy Sector	SHORT	
24	BBVA, CaixaBank, Banco Sabadell: Layoffs and Restructuring	LONG	LONG
25	Inditex, Acerinox: Market Performance and Strategic Developments in the Post-Covid Context	SHORT	SHORT

Note: Mapping of embeddings-based KMeans clusters to their Trading Signal (LONG/SHORT) *for the two proposed cluster-selection algorithms (Greedy and Stable). The Greedy algorithm longs (shorts) clusters that maximize (minimize) the cluster-average-SR in the validation sample subject to a positivity (negativity) constraint, while the Stable algorithm longs (shorts) clusters that minimize the rank difference between the training and validation rankings of the cluster-average-SR's subject to a positivity (negativity) constraint, which is now imposed on both sample splits. In both algorithms, the cardinality of each leg is upper-bounded by a hyperparameter θ. Cluster labels are proposed based on the articles they pool.*

TABLE 4: Mapping of LLM-based clusters to Trading Signals

	Cluster	Greedy	Stable
0	(demand, minor, positive)		
1	(demand, minor, negative)		SHORT
2	(demand, major, positive)	SHORT	SHORT
3	(demand, major, negative)	LONG	LONG
4	(supply, minor, positive)	LONG	
5	(supply, minor, negative)	SHORT	
6	(supply, major, positive)	LONG	
7	(supply, major, negative)	SHORT	
8	(financial, minor, positive)	LONG	LONG
9	(financial, minor, negative)		SHORT
10	(financial, major, positive)	LONG	
11	(financial, major, negative)	SHORT	
12	(technology, minor, positive)	LONG	
13	(technology, minor, negative)		
14	(technology, major, positive)	SHORT	
15	(technology, major, negative)		
16	(policy, minor, positive)	SHORT	SHORT
17	(policy, minor, negative)	SHORT	SHORT
18	(policy, major, positive)	SHORT	SHORT
19	(policy, major, negative)	SHORT	SHORT

Note: Mapping of LLM-based clusters to their Trading Signal (LONG/SHORT) *for the two proposed cluster-selection algorithms (Greedy and Stable). The Greedy algorithm longs (shorts) clusters that maximize (minimize) the cluster-average-SR in the validation sample subject to a positivity (negativity) constraint, while the Stable algorithm longs (shorts) clusters that minimize the rank difference between the training and validation rankings of the cluster-average-SR's subject to a positivity (negativity) constraint, which is now imposed on both sample splits. In both algorithms, the cardinality of each leg is upper-bounded by a hyperparameter θ. Each cluster corresponds to a type of news-implied firm-specific shock identified by our LLM according to the function calling schema.*

TABLE 5: Portfolio Statistics Comparison: KMeans vs LLM Clustering

(A) Panel A: Statistics of $\mathcal{P}_{\text{KMeans}}$

Split	Algo.	Cum. Ret.	Avg. Ret.	St. Dev.	Sharpe Ratio	Sortino Ratio	Max. DD	Calmar Ratio	Skew.	Exc. Kurt.	VaR 95%	CVaR 95%
All	*Greedy*	1.070	5.3	9.7	0.5	0.6	-6.9	0.8	-0.45	4.04	-13.7	-22.9
	Stable	1.489	35.8	16.8	1.8	2.2	-7.6	4.7	0.19	5.08	-22.6	-36.1
Train	*Greedy*	0.959	-6.2	11.7	-0.5	-0.5	-6.9	-0.9	-0.52	2.72	-18.3	-28.5
	Stable	1.250	40.4	19.6	1.7	2.0	-7.6	5.3	-0.22	3.24	-29.3	-43.1
Validation	*Greedy*	1.088	26.8	7.3	3.3	3.7	-3.5	7.8	-0.47	1.17	-9.5	-15.9
	Stable	1.149	47.6	13.3	2.9	3.5	-3.6	13.1	-0.19	1.76	-18.3	-28.1
Test	*Greedy*	1.014	4.9	6.9	0.7	1.0	-3.6	1.4	1.85	5.50	-7.8	-9.7
	Stable	1.008	2.9	14.3	0.2	0.3	-4.6	0.6	2.46	14.57	-18.9	-26.8

(B) Panel B: Statistics of $\mathcal{P}_{\text{LLM}}$

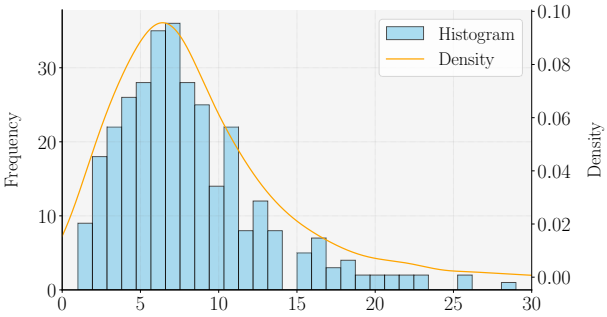
Split	Algo.	Cum. Ret.	Avg. Ret.	St. Dev.	Sharpe Ratio	Sortino Ratio	Max. DD	Calmar Ratio	Skew.	Exc. Kurt.	VaR 95%	CVaR 95%
All	*Greedy*	1.310	23.1	9.6	2.2	2.9	-6.3	3.7	1.47	9.93	-13.6	-18.9
	Stable	1.365	27.0	8.6	2.8	3.4	-5.9	4.6	0.28	2.24	-11.9	-16.9
Train	*Greedy*	1.112	17.6	11.4	1.4	1.9	-6.3	2.8	1.65	9.00	-15.7	-21.0
	Stable	1.177	28.3	9.9	2.5	3.0	-5.9	4.8	0.16	1.71	-13.5	-19.6
Validation	*Greedy*	1.091	28.0	8.2	3.0	4.0	-3.1	9.1	0.14	1.37	-10.6	-16.8
	Stable	1.048	14.2	7.0	1.9	2.1	-1.9	7.4	0.25	1.37	-11.1	-14.7
Test	*Greedy*	1.084	30.8	6.2	4.3	6.0	-1.5	21.0	1.49	8.30	-6.9	-9.9
	Stable	1.100	37.2	7.1	4.4	7.2	-1.1	34.5	0.84	1.95	-9.5	-11.3

Note: Portfolio statistics of trading strategies based on clusters obtained from KMeans (Panel A) and LLM (Panel B) approaches. The statistics provided include performance metrics (Cumulative Return, Average Return (%)), risk measures (Standard Deviation (%), Maximum Drawdown (%), Value at Risk (%), Conditional Value at Risk (%)), risk-adjusted performance ratios (Sharpe Ratio, Sortino Ratio, Calmar Ratio), and return distribution characteristics (Skewness, Excess Kurtosis). These statistics are provided for both cluster-selection algorithms: Greedy and Stable. Except for the Cumulative Return, all returns are annualized. The Sharpe Ratio is computed using the daily returns, assuming 252 trading days in a year. The Sortino Ratio is calculated using the daily downside returns. The Maximum Drawdown is the maximum loss from a peak to a trough. The Calmar Ratio is the ratio of the annualized return to the maximum drawdown. Skewness measures the asymmetry of the return distribution, while Kurtosis quantifies the tails' thickness. The Value at Risk (VaR) and Conditional Value at Risk (CVaR) are calculated at a 95% confidence level. The Greedy algorithm longs (shorts) clusters that maximize (minimize) the cluster-average-SR in the validation sample subject to a positivity (negativity) constraint, while the Stable algorithm longs (shorts) clusters that minimize the rank difference between the training and validation rankings of the cluster-average-SR's subject to a positivity (negativity) constraint, which is now imposed on both sample splits. In both algorithms, the cardinality of each leg is upper-bounded by a hyperparameter θ. The holding period of the beta-neutral positions is set to $L = 4$ trading days for both approaches. The number of traded clusters is $\theta = 0.5k = 13$ for KMeans ($k^* = 26$ clusters) and $\theta = 0.5k = 10$ for LLM ($k^* = 20$ clusters). The selection criteria for these hyperparameters (L, θ) is based on maximizing the Sharpe Ratios of the train and validation samples.

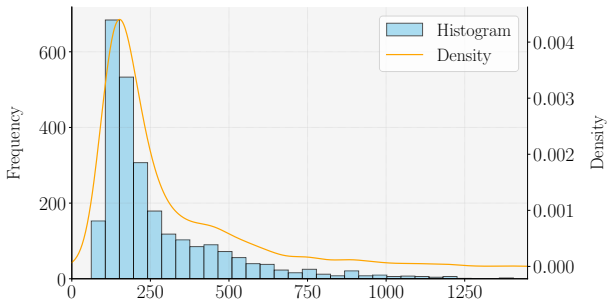
FIGURE 1: Word Cloud of all the dataset

Note: This Word Cloud visualizes the most frequent words in our dataset of Spanish business news articles. Larger words correspond to higher frequencies. The color of the words is purely for visual differentiation and holds no additional meaning. The most prominent words include "empresa" (firm), "compañía" (company), and "españa" (Spain), reinforcing that the dataset primarily comprises Spanish business news, with a prevalence of technical terms such as "beneficio neto" (net profit), "precio objetivo" (target price), "proyecto" (project), and "operación" (operation).

FIGURE 2: Histogram of # News Articles per Day and # Words per Article



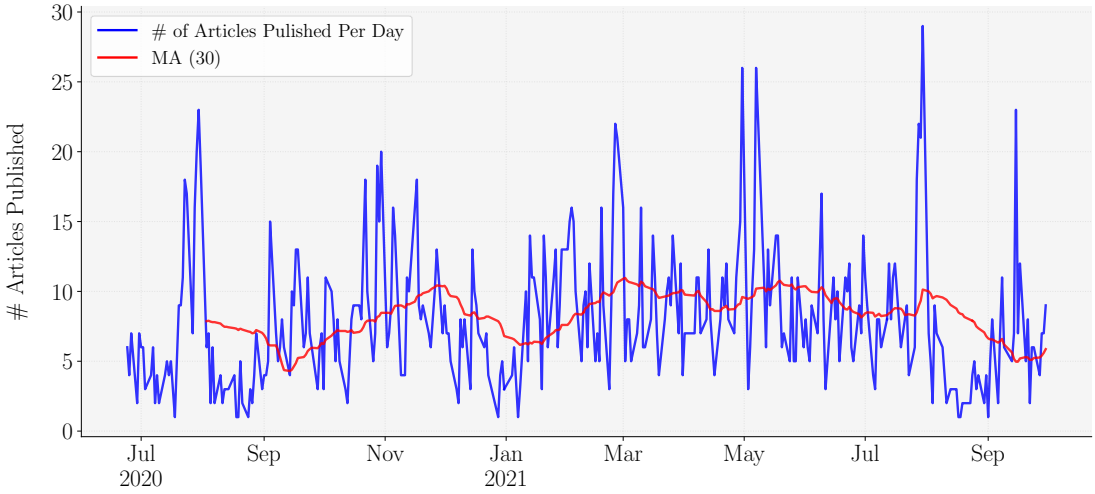
(A) Number of News Articles per Day



(B) Number of Words per Article

Note: Panel (a) displays the distribution of the number of news articles published per day, with most days having between 5 and 10 articles. Panel (b) shows the distribution of the number of words per article, where the majority are between 70 and 280 words, suggesting concise reporting. However, the long right tail indicates instances of more comprehensive coverage.

FIGURE 3: Time Series of Number of Articles per Day and 30-Period Moving Average

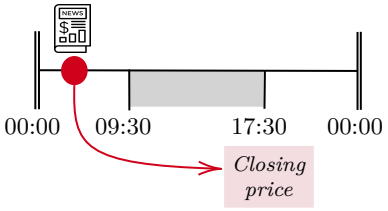


Note: The time series shows the daily number of news articles published, characterized by significant variability with occasional sharp spikes. The 30-day moving average smooths these fluctuations, revealing an average publication rate of 5 to 10 articles per day.

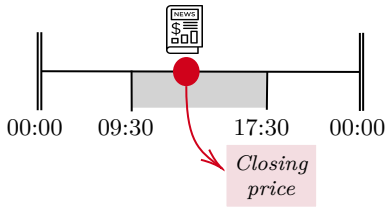
FIGURE 4: Imputation of the effective treatment date ($\tilde{d}_0^i$)

Case 1: Treatment date is the same as the publication date; $\tilde{d}_0^i = d_0^i$

Case 1a: News article published in a trading date and before the market opens; $\tilde{d}_0^i \in \tilde{\mathfrak{d}} \ \wedge \ t_0^i < 09{:}30$

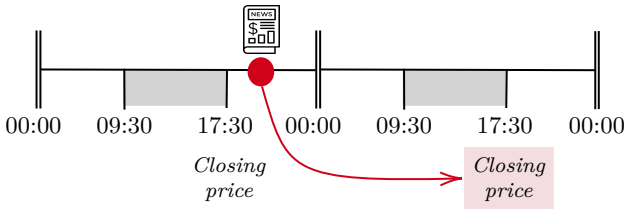


Case 1b: News article published in a trading date and during market hours; $\tilde{d}_0^i \in \tilde{\mathfrak{d}} \wedge t_0^i \in [09{:}30, 17{:}30]$

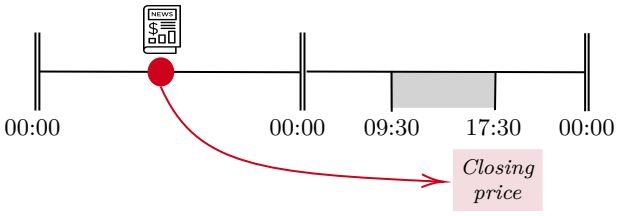


Case 2: Treatment date is the next closest trading day to publication; $\tilde{d}_0^i = \Lambda(d_0^i)$

Case 2a: News article published in a trading day but after the market is closed for that day; $d_0^i \in \tilde{\mathfrak{d}} \ \wedge \ t_0^i > 17{:}30$

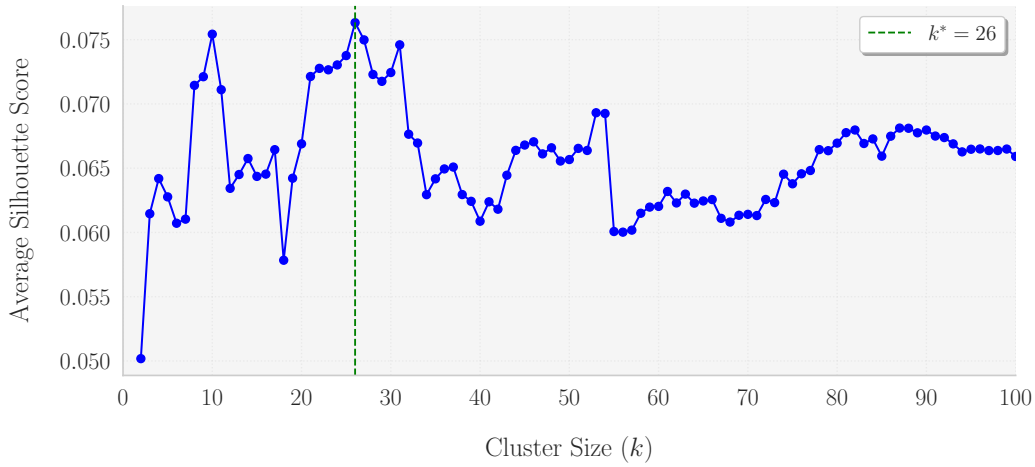


Case 2b: News article published in a non-trading day; $d_0^i \notin \tilde{\mathfrak{d}}$



Note: This figure illustrates how we determine the effective treatment date for news articles based on their publication timing relative to market hours. The Spanish stock market operates from 09:30 to 17:30 on trading days. News published during a trading day during or before trading hours affects stock prices on the same day (Cases 1a and 1b), while news published after market close or on non-trading days affects prices on the next available trading day (Cases 2a and 2b). This temporal mapping ensures we correctly align news publication with the first opportunity for market reaction.

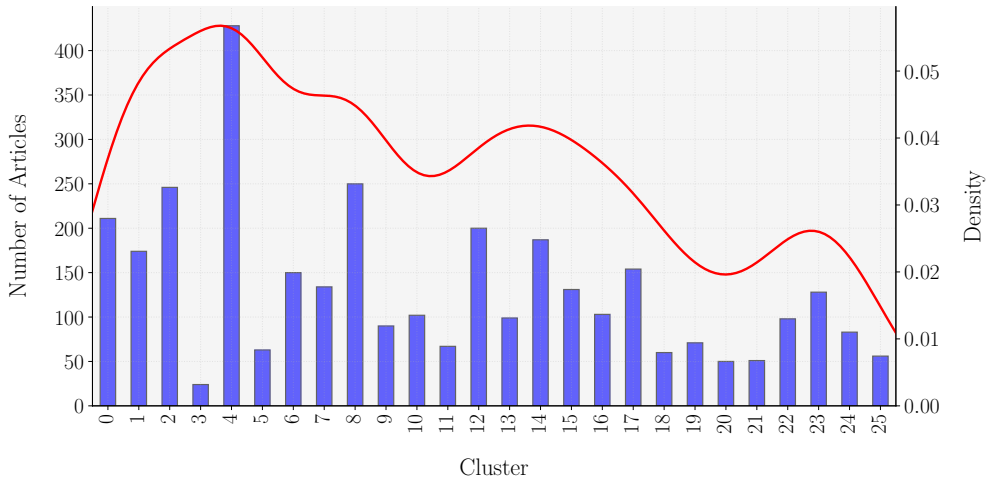
FIGURE 5: Average Silhouette Scores in the Training data



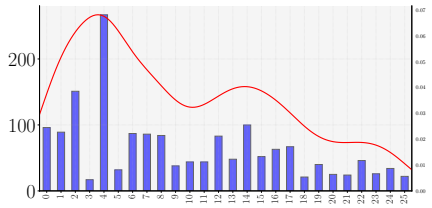
Note: The plot presents the average silhouette scores calculated on the training data $\mathcal{D}^{tr}$ for various cluster sizes k ranging from 2 to 100. The silhouette score measures how well data points fit within their assigned cluster by comparing intra-cluster cohesion with inter-cluster separation. A higher silhouette score (closer to +1) indicates better-defined clusters. The optimal number of clusters, $k^* = 26$, which maximizes the average silhouette score, is marked by a vertical dashed green line.

FIGURE 6: Distribution of articles through KMeans clusters

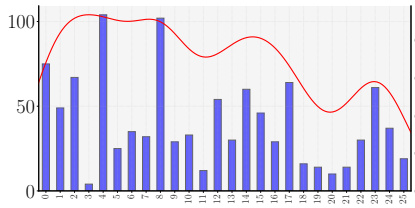
(A) All data ($\mathcal{D}$)



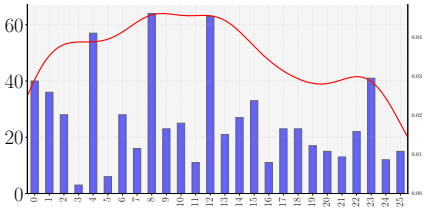
(B) Training data ($\mathcal{D}^{tr}$)



(C) Validation data ($\mathcal{D}^{val}$)



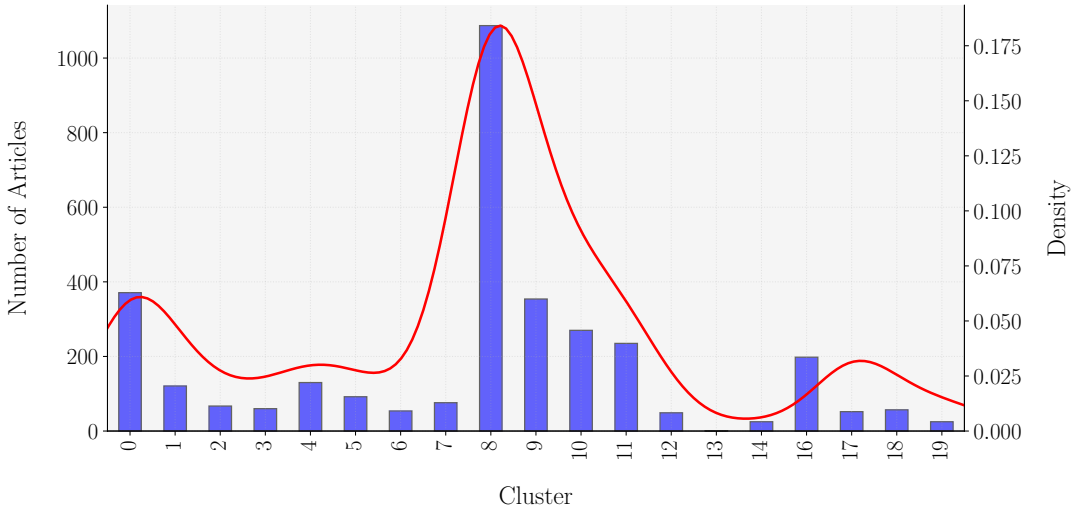
(D) Test data ($\mathcal{D}^{test}$)



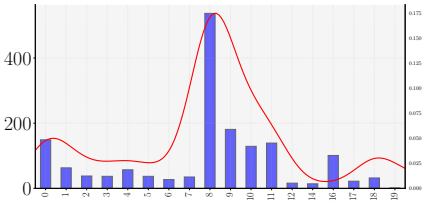
Note: This figure presents the distribution of articles across the $k^* = 26$ clusters, where the centroids were determined by applying the KMeans algorithm to the article embeddings from the training data. Panel (A) *shows the distribution for the entire dataset ($\mathcal{D}$), while Panels* (B), (C), *and* (D) *illustrate the distributions for the training ($\mathcal{D}^{tr}$), validation ($\mathcal{D}^{val}$), and test ($\mathcal{D}^{test}$) datasets, respectively. The differences in distribution across splits suggest some temporal instability in the clustering results.*

FIGURE 7: Distribution of articles through LLM clusters

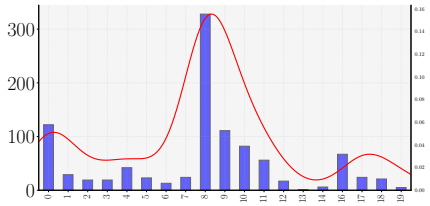
(A) All data ($\mathcal{D}$)



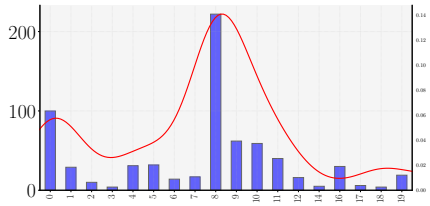
(B) Training data ($\mathcal{D}^{tr}$)



(C) Validation data ($\mathcal{D}^{val}$)



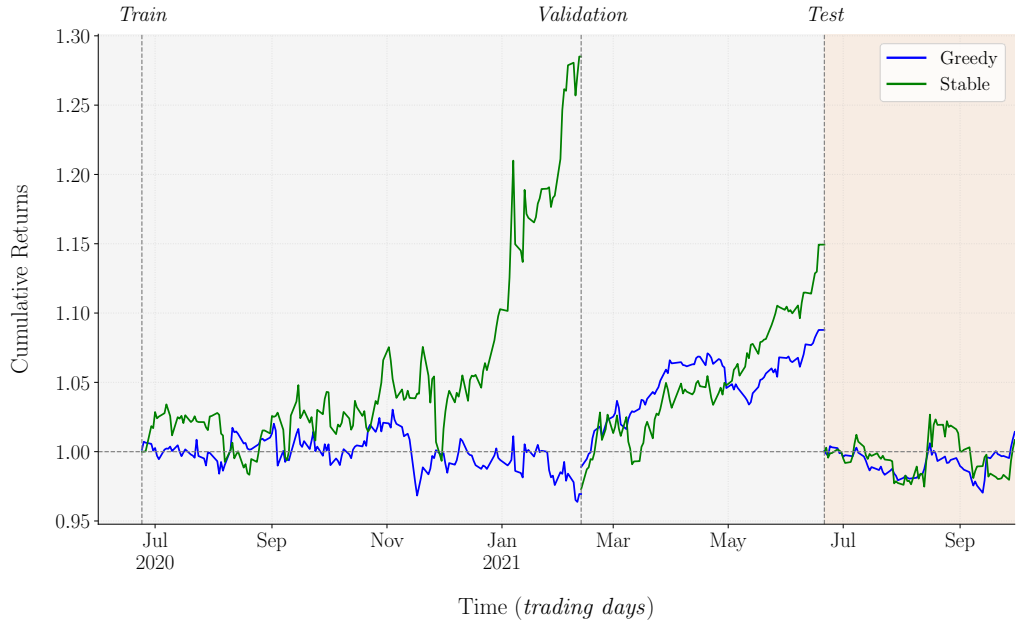
(D) Test data ($\mathcal{D}^{test}$)



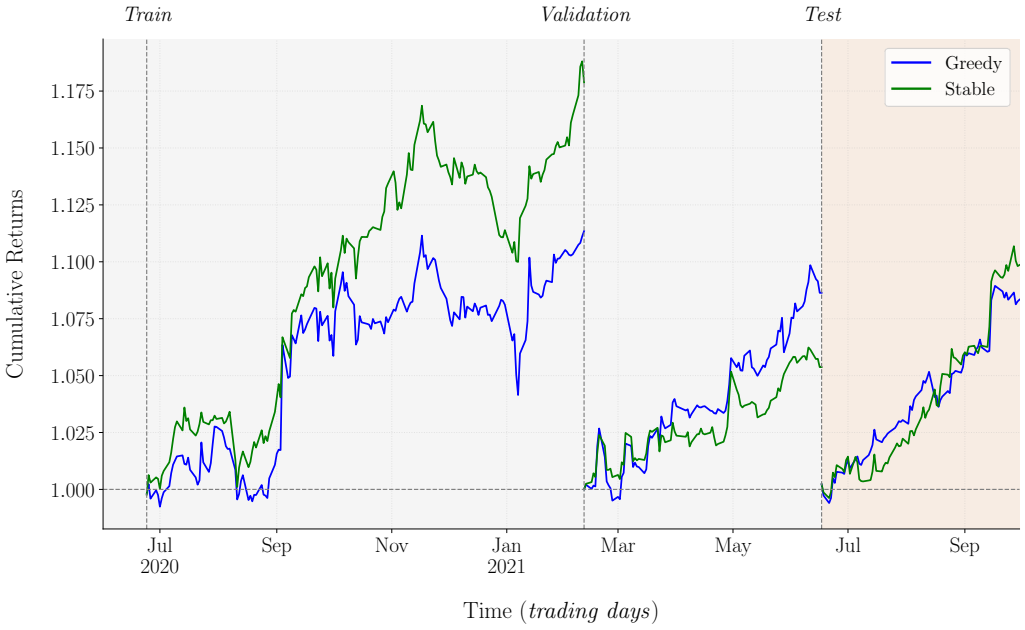
Note: This figure presents the distribution of news articles across clusters derived using an LLM-based approach. The upper plot shows the distribution for the entire dataset ($\mathcal{D}$), while the lower plots display the distributions for the training ($\mathcal{D}^{tr}$), validation ($\mathcal{D}^{val}$), and test ($\mathcal{D}^{test}$) datasets. Clusters 8, 9, 10, and 11, which capture financial events or shocks, dominate the distribution, with cluster 8 (financial, minor, positive) representing approximately one-third of the dataset. This cluster includes articles related to financial reports with mildly positive outcomes, potentially offering insight for long trading signals. Unlike KMeans clustering with embeddings, this LLM-based clustering shows stable distributions across data splits, highlighting the robustness of this method over time.

FIGURE 8: Comparison of Cumulative Gross Returns across Clustering Approaches

(A) Panel A: Cumulative Gross Returns of $\mathcal{P}_{\text{KMeans}}$



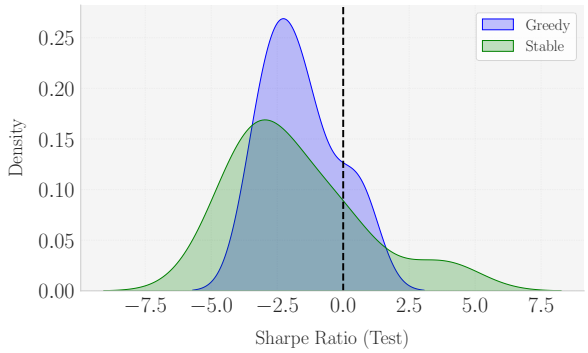
(B) Panel B: Cumulative Gross Returns of $\mathcal{P}_{\text{LLM}}$



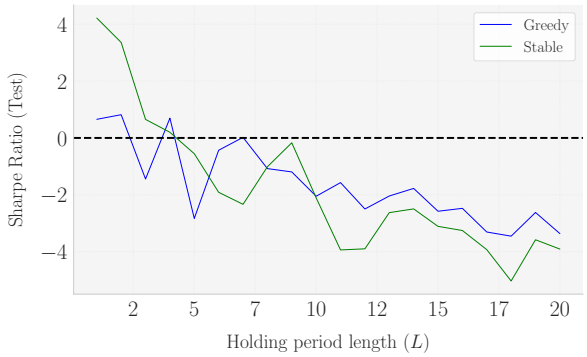
Note: This figure presents the cumulative gross returns of trading strategies based on KMeans clustering (Panel A) and LLM clustering (Panel B) across different data splits. For both approaches, the holding period of the beta-neutral strategies is set to $L = 4$ trading days. The number of traded clusters differs between approaches: $\theta = \lfloor 0.5k \rfloor = 13$ for KMeans ($k^* = 26$ clusters) and $\theta = \lfloor 0.5k \rfloor = 10$ for LLM ($k = 20$ clusters). The selection criteria for these parameters is based on maximizing the Sharpe Ratios of the train and validation samples. The Test split is higlhighted with a yellow background.

FIGURE 9: Sensitivity of $SR^{\mathcal{P}^{test}}$ to the holding window length (L)

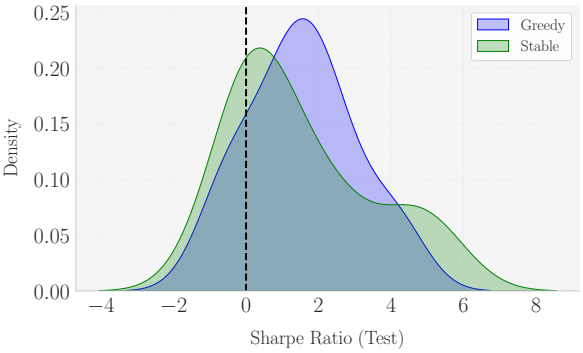
(A) **KMeans**: Distribution of $SR^{\mathcal{P}^{test}}(L)$



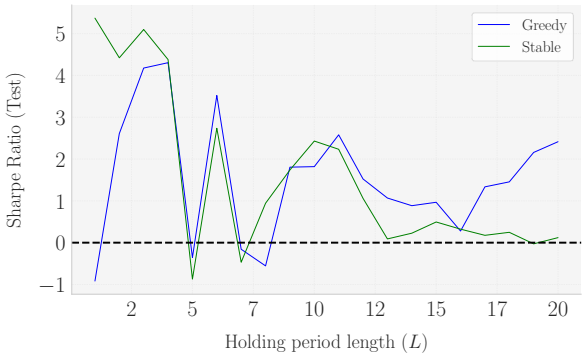
(B) **KMeans**: Series of $SR^{\mathcal{P}^{test}}(L)$



(C) **LLM**: Distribution of $SR^{\mathcal{P}^{test}}(L)$



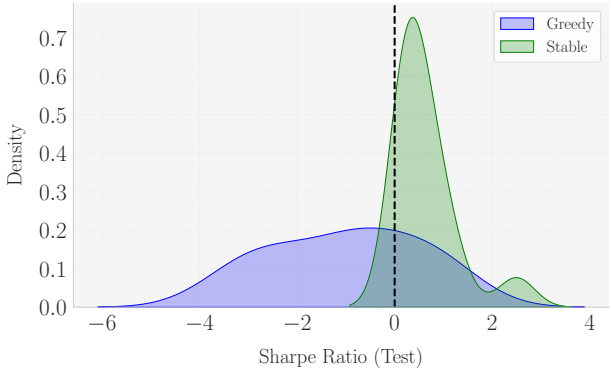
(D) **LLM**: Series of $SR^{\mathcal{P}^{test}}(L)$



Note: This figure examines the sensitivity of the Sharpe Ratios ($SR^{\mathcal{P}^{test}}$) of the test portfolio to changes in the holding window length (L), with θ fixed at $\lfloor 0.5k \rfloor$. Panels (A) *and* (B) *display the distribution and time series of $SR^{\mathcal{P}^{test}}(L)$ for KMeans clustering, respectively, while Panels* (C) *and* (D) *present the same for the LLM-based clustering. The left-hand panels show the skewness of the distributions: KMeans clustering results in a left-skewed distribution of Sharpe Ratios, whereas the LLM-based approach yields a right-skewed distribution, indicating higher profitability. The right-hand panels highlight that KMeans clustering only produces positive Sharpe Ratios for very short holding periods, whereas the LLM-based clustering shows more consistent positive performance across a wider range of L values, though with some variability.*

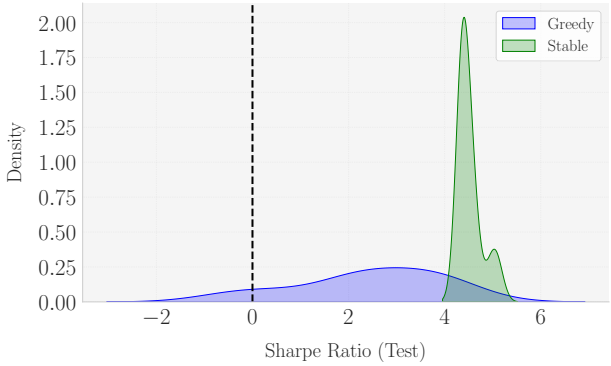
FIGURE 10: Sensitivity of $SR^{\mathcal{P}^{test}}$ to the upper bound on the number of traded clusters on each side (θ)

(A) **KMeans**: Distribution of $SR^{\mathcal{P}^{test}}(\theta)$



(B) **KMeans**: Series of $SR^{\mathcal{P}^{test}}(\theta)$

(C) **LLM**: Distribution of $SR^{\mathcal{P}^{test}}(\theta)$



(D) **LLM**: Series of $SR^{\mathcal{P}^{test}}(\theta)$

Note: This figure displays the sensitivity of the Sharpe Ratios ($SR^{\mathcal{P}^{test}}$) to variations in the upper bound on the number of traded clusters (θ), with L fixed at 4. Panels (A) *and* (B) *show the distribution and series of $SR^{\mathcal{P}^{test}}(\theta)$ for KMeans clustering, respectively, while Panels* (C) *and* (D) *illustrate the same for LLM-based clustering. For KMeans, the results are mixed: the Stable algorithm generates positive Sharpe Ratios for low θ values, whereas the Greedy algorithm performs better with high θ values, indicating sensitivity and instability. In contrast, the LLM-based clustering shows a more consistent pattern, with a concentration of positive Sharpe Ratios across a broader range of θ values, suggesting greater robustness and stability in the trading strategy.*

A. Appendix

A.1 KMeans Algorithm

Algorithm 1. KMeans Clustering Algorithm

Input: Embedding vectors $\{\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^N\}$, number of clusters k
Output: Cluster assignments $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k\}$, centroids $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$
Initialize centroids $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$ randomly
repeat
    Assignment Step:
    for each vector $\mathbf{e}^i$ **do**
        Assign $\mathbf{e}^i$ to the nearest centroid:

$$g = \arg \min_{\ell \in \{1,\ldots,k\}} \|\mathbf{e}^i - \mathbf{c}_\ell\|_2^2$$

        Update cluster assignments: $\mathcal{D}_g \leftarrow \mathcal{D}_g \cup \{i\}$
    end for
    Update Step:
    for each cluster $\mathcal{D}_g$ **do**
        Recalculate centroid $\mathbf{c}_g$:

$$\mathbf{c}_g = \frac{1}{|\mathcal{D}_g|} \sum_{i \in \mathcal{D}_g} \mathbf{e}^i$$

    end for
until cluster assignments no longer change
Return cluster assignments $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k\}$ and centroids $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$

A.2 Hyperparameter Choice Justification

Our hyperparameters are L and θ. Recall that L denotes the number of trading days over which we hold the positions in the beta-neutral strategy, while θ represents the upper bound on each side (long and short) for the amount of clusters we select for the trading strategy. The specific choice of hyperparameters we made for the results presented in the paper were:

$$L = 4$$
$$\theta = \lfloor 0.5k \rfloor$$

where k represents the number of clusters (26 for KMeans clustering, and 20 for LLM clustering). This choice is not arbitrary nor opportunistic. Instead, it results from the maximization of the Sharpe Ratio of the portfolio in the train and validation samples for both KMeans and LLM clustering. This choice procedure is completely based on *in-sample* criteria and it prevents lookahead bias. The justification for such choices is made below.

A.2.1 KMeans Clustering

In Figure A1 we can see that a choice of $L = 4$ in the training and validation splits generates the most stable Sharpe Ratio. Namely, In the train set (Figure A1a), it makes more sense to choose low values of L (less than 4) to maximize the SR. However, in the validation set (Figure A1b), it makes more sense to choose higher values of L. The choice of $L = 4$ represents a balanced compromise, providing a stable Sharpe Ratio profile across both splits, ensuring consistent in-sample performance.

[Insert Figure A1 about here]

On the other hand, the choice of $\theta = \lfloor 0.5 \cdot 26 \rfloor = 13$ is a choice that pursues stability in the Sharpe Ratio of the train and validation portfolios. As we can see from Figure A2, the Sharpe Ratios tend to converge to the highest and most stable value when we choose the highest possible value of θ.

[Insert Figure A2 about here]

A.2.2 LLM Clustering

Following a similar logic as below, the choice of $L = 4$ sets a consensus between the maximization of $SR^{\mathcal{P}^{tr}}$ and $SR^{\mathcal{P}^{val}}$. That is, maximizing $SR^{\mathcal{P}^{tr}}$ requires lower holding period lengths (the maximizer is $L = 4$), while maximizing $SR^{\mathcal{P}^{val}}$ requires increasing the window length. Among this contradiction, from Figure A3 it follows that $L = 4$ stands as a perfect choice to balance the maximization requirements in both samples, generating a stable choice for the holding period window length.

[Insert Figure A3 about here]

Finally, the same conclusion as in KMeans applies here. By selecting $\theta = \lfloor 0.5 \cdot 20 \rfloor = 10$, we get a stable Sharpe Ratio. Even though we observe that $SR^{\mathcal{P}^{tr}}(L)$ falls momentarily at $\theta = 10$ for the Greedy algorithm, it still constitutes a good choice. Conversely, at $\theta = 10$ the greedy algorithm sees a jump in $SR^{\mathcal{P}^{val}}(L)$ (see Figure A4). All in all, we can easily conclude that $\theta = \lfloor 0.5k \rfloor$ arises as a good hyperpamrameter choice also for LLM clustering.

[Insert Figure A4 about here]

A.3 Cluster-Average Sharpe Ratios

The distribution of cluster-average Sharpe Ratios across different clusters reveals distinct patterns between KMeans and LLM-based clustering approaches, as illustrated in Figure A5

Panel A presents the results for KMeans clustering, where we observe remarkably consistent distributional patterns across all three data splits. The distributions are approximately symmetric around zero, with the majority of Sharpe ratios falling within the $[-5, 5]$ range. The training set exhibits the highest density peak (approximately 0.17), followed closely by the test set, while the validation set shows a slightly lower peak density of about 0.125. Notable in the validation set are small secondary peaks at the tails (around ± 15), suggesting the presence of a few clusters with extreme performance characteristics. This consistency across splits suggests that the KMeans clustering approach produces stable performance groupings.

Panel B displays the results for LLM-based clustering, revealing more heterogeneous distributions across the splits. The validation set demonstrates a pronounced peak near zero with a maximum density of 0.2, indicating strong concentration of performance in this region. In contrast, the training set exhibits a markedly different pattern, with a flatter, more dispersed distribution extending from -20 to $+20$, suggesting greater performance variability across clusters. The test set presents an intermediate case, with moderate concentration around zero but maintaining significant mass in the positive region. This heterogeneity across splits might indicate that the LLM-based clustering captures more nuanced and potentially time-varying patterns in the underlying data.

The contrasting patterns between the two clustering approaches suggest different strengths: KMeans provides more stable and consistent performance groupings, while LLM-based clustering potentially captures more complex relationships, albeit with greater variability across different data splits.

[Insert Figure A5 about here]

A.4 Optimal Cluster Selection Algorithms

Algorithm 2. GREEDY SELECTION | Top average Sharpe Ratio in Validation Set

Input: Set of clusters $\mathcal{G} = \{1, 2, \ldots, k^*\}$, Sharpe Ratios in the validation sample $\{SR_L^{(i,j)}\}_{(i,j)\in\mathcal{B}^{val}}$, maximum number of traded clusters $\theta \in \mathbb{N}$ (usually, $\theta \propto k^*$)

Output: Set of long-traded clusters $\mathcal{G}_\theta^+$ and set of short-traded clusters $\mathcal{G}_\theta^-$

Step #1: Compute Cluster Average Sharpe Ratios in Validation Set

for each $g \in \mathcal{G}$ **do**

  Compute average Sharpe Ratio $\overline{SR}_g^{val} \leftarrow \frac{1}{|\mathcal{B}_g^{val}|} \sum_{(i,j)\in\mathcal{B}_g^{val}} SR_L^{(i,j)}$

end for

Step #2: Identify Positive and Negative Sharpe Ratio Clusters

Define $\mathcal{G}_{SR^+}^{val} \leftarrow \{g \in \mathcal{G} \mid \overline{SR}_g^{val} > 0\}$

Define $\mathcal{G}_{SR^-}^{val} \leftarrow \{g \in \mathcal{G} \mid \overline{SR}_g^{val} < 0\}$

Step #3: Rank Clusters by Average Sharpe Ratio in the Validation Set

for each $g \in \mathcal{G}$ **do**

  Rank the average Sharpe Ratio $\mathfrak{R}_g^{val} \leftarrow \sum_{h\in\mathcal{G}} \mathbf{1}\left(\overline{SR}_h^{val} \geq \overline{SR}_g^{val}\right)$

end for

Step #4: Select Top θ Clusters

Define $\theta^+ \leftarrow \min(\theta, |\mathcal{G}_{SR^+}^{val}|)$; $\mathcal{G}_\theta^+ \leftarrow \{g \in \mathcal{G} \mid 1 \leq \mathfrak{R}_g^{val} \leq \theta^+\}$

Define $\theta^- \leftarrow \min(\theta, |\mathcal{G}_{SR^-}^{val}|)$; $\mathcal{G}_\theta^- \leftarrow \{g \in \mathcal{G} \mid k^* - \theta^- < \mathfrak{R}_g^{val} \leq k^*\}$

Return Long-traded clusters $\mathcal{G}_\theta^+$, Short-traded clusters $\mathcal{G}_\theta^-$

Algorithm 3. RANK STABILITY | Minimal Rank Difference between Train & Validation Sets

Input: Set of clusters $\mathcal{G} = \{1, 2, \ldots, k^*\}$, Sharpe Ratios in the training and validation sample $\{SR_L^{(i,j)}\}_{(i,j)\in\mathcal{B}^{tr}}$ and $\{SR_L^{(i,j)}\}_{(i,j)\in\mathcal{B}^{val}}$, maximum number of traded clusters θ

Output: Set of long-traded clusters $\mathcal{G}_\theta^+$ and set of short-traded clusters $\mathcal{G}_\theta^-$

Step #1: Compute Cluster Average Sharpe Ratios in Training & Validation Set

for each $g \in \mathcal{G}$ **do**

  Compute average Sharpe Ratio in $\mathcal{B}^{tr}$: $\overline{SR}_g^{tr} \leftarrow \frac{1}{|\mathcal{B}_g^{tr}|}\sum_{(i,j)\in\mathcal{B}_g^{tr}} SR_L^{(i,j)}$

  Compute average Sharpe Ratio in $\mathcal{B}^{val}$: $\overline{SR}_g^{val} \leftarrow \frac{1}{|\mathcal{B}_g^{val}|}\sum_{(i,j)\in\mathcal{B}_g^{val}} SR_L^{(i,j)}$

end for

Step #2: Rank Clusters

for each $g \in \mathcal{G}$ **do**

  Rank the average Sharpe Ratios in $\mathcal{B}^{tr}$: $\mathfrak{R}_g^{tr} \leftarrow \sum_{h\in\mathcal{G}} \mathbf{1}\left(\overline{SR}_h^{tr} \geq \overline{SR}_g^{tr}\right)$

  Rank the average Sharpe Ratios in $\mathcal{B}^{val}$: $\mathfrak{R}_g^{val} \leftarrow \sum_{h\in\mathcal{G}} \mathbf{1}\left(\overline{SR}_h^{val} \geq \overline{SR}_g^{val}\right)$

end for

Step #3: Calculate Rank Differences

for each $g \in \mathcal{G}$ **do**

  Calculate rank difference $\delta_g \leftarrow |\mathfrak{R}_g^{tr} - \mathfrak{R}_g^{val}|$

end for

Step #4: Select Top θ Clusters with Smallest Rank Differences

for each $g \in \mathcal{G}$ **do**

  Rank the rank difference : $\mathfrak{R}(\delta_g) \leftarrow \sum_{h\in\mathcal{G}} \mathbf{1}\left(\delta_g \geq \delta_h\right)$

end for

Select top 2θ clusters with smallest δ_g: $\mathcal{G}_\theta = \{g \in \mathcal{G} \mid 1 \leq \mathfrak{R}(\delta_g) \leq 2\theta\}$

 # Step 5: Determine Long and Short Positions

Define $\mathcal{G}_\theta^+ = \{g \in \mathcal{G}_\theta \mid \overline{SR}_g^{tr} > 0 \text{ and } \overline{SR}_g^{val} > 0\}$

Define $\mathcal{G}_\theta^- = \{g \in \mathcal{G}_\theta \mid \overline{SR}_g^{tr} < 0 \text{ and } \overline{SR}_g^{val} < 0\}$

Return Long-traded clusters $\mathcal{G}_\theta^+$, Short-traded clusters $\mathcal{G}_\theta^-$

A.5 Sample of articles for each cluster

Table A1: KMeans clustering. Proposed name for the clusters and sample of 3 articles for each cluster.

#	Title	Articles
0	Miscellaneous (Colonial, Acciona, Amadeus, Grifols, Endesa, IAG, Bankinter...)	• Colonial forecasts rental income of EUR338m in 2020 • Acciona's asset sales will allow it to grow in renewables • Sabadell recommends selling Amadeus shares due to worse sales forecast.
1	Quarterly & Semi-Annual Earnings Reports	• Enagás 1H net profit falls 9.8% due to lower income and extraordinary items. • Iberdrola: Net profit of EUR1.025m in Q1 • Santander almost quintuples Q1 profit due to absence of Covid provisions.
2	BBVA & Sabadell: Financial Performance & Strategic Movements	• Interest rate hike in Turkey favors BBVA's net interest margin • Sabadell reorganizes business in Spain following the arrival of the new CEO. • Fitch downgrades Banco Sabadell's rating one notch to low grade.
3	Telefónica & Cellnex: Telecommunications Tower Sales & Market Dynamics	• Telefónica shares soar after selling towers of its subsidiary in Europe and Latin America. • Telefónica hires Goldman Sachs to sell its British tower business • Dutch Competition Authority authorizes Cellnex to integrate 3,150 Deutsche Telekom towers.
4	CaixaBank: Mergers and Strategic Moves in the Banking Sector	• CaixaBank and Bankia approve their merger project • CaixaBank closes its first issuance of green bonds in pounds for 500 million • CaixaBank-Bankia merger could generate EUR500m in savings
5	Telefónica, Indra, & MásMóvil: Regulatory and Strategic Moves in Telecom	• Indra to partner with Telefónica in the deployment of fiber optics in Germany. • Telefónica launches a buyback offer for its hybrid bonds of EUR1.000m. • EU refers Liberty Global and Telefónica agreement to UK regulator
6	Siemens Gamesa: Supply Agreements, Profitability Targets in Renewable Energy	• Siemens Gamesa will supply turbines to Elawan's 150 MW wind farm in Spain. • Siemens Gamesa lowers its profitability target for 2021. • Siemens Gamesa will supply 160 MW for the largest wind farm in the Philippines.
7	Cellnex: Strategic Acquisitions and Financial Moves in Telecom Infrastructure	• Cellnex launches a EUR1.850m debt issue • Cellnex agrees to buy 10,500 telecommunications towers in France for EUR5.200m • Benetton family sells 2.5% of Cellnex to Singapore sovereign fund
8	Acciona, Endesa, Enagás & Naturgy: Strategic Moves & Regulatory Developments in the Energy Sector	• Naturgy and Enagás study project to produce green hydrogen in Asturias • Break of ties between Algeria and Morocco may damage gas flow to Spain • Acciona: Energy business IPO on track for 1H
9	Repsol: Strategic Moves and Challenges in the Energy Sector	• Repsol to produce green hydrogen at Petronor refinery in 2022 • Repsol and Talgo to jointly promote the creation of renewable hydrogen trains • Repsol gains access to a portfolio of renewable assets in Chile through a joint venture
10	Ferrovial, Acciona: Strategic Expansions and Financial Maneuvers in Infrastructure	• Ferrovial closes the sale of Broadspectrum to Ventia for EUR291m • Acciona awarded the construction of 2 roads in Poland for EUR642m • Renfe awards on-board services contract to Ferrovial for EUR272m
11	Solaria: Strategic Moves and Market Challenges in Renewable Energy	• Solaria invests EUR220m in Europe's largest photovoltaic park. • Solaria will supply energy to Shell and Axpo with Europe's largest photovoltaic plant • Goldman Sachs downgrades Solaria recommendation after stock rise.
12	Iberdrola: Strategic Collaborations and Renewable Energy Developments	• Iberdrola will build a self-consumption plant for Lactalis factory in Spain. • Iberdrola and Mapfre launch a renewable energy co-investment vehicle in Spain. • Iberdrola partners with Mitsubishi to decarbonize the industry.

13	IAG: Financial Performance	• IAG Q3 results worse than expected • IAG burns cash faster than anticipated • IAG stock may be pricing in a second capital increase
14	Santander & CaixaBank: Financial Moves and Sustainability Initiatives	• CaixaBank mobilizes EUR12.000m in sustainable financing in the first 9 months of 2020. • EIB and Banco Santander will inject EUR587m into Portuguese SMEs. • Banco Santander, leader in renewable project financing in 2020.
15	ACS & Acciona: Strategic Movements and Infrastructure Projects	• ACS and Acciona win contracts for new Australian airport worth EUR164m. • Acciona awarded 3 contracts to operate wastewater treatment plants in Sardinia for EUR210m. • ACS expects net profit to grow by around 30% in 2021
16	Telefónica: Financial Performance and Strategic Moves	• Reduction in Telefónica's debt will improve analysts' perception • Telefónica's profit more than doubles in Q1 due to lower financial expenses. • Telefónica, América Móvil and TIM buy the mobile network of Brazil's Oi.
17	Meliá and Spanish Tourism Sector: Challenges Amidst the Pandemic	• Meliá: Spanish hotel sector faces another uncertain summer with cautious optimism. • Meliá claims EUR116m from the Spanish government for pandemic-related damages. • Meliá: Local Covid-19 lockdowns will continue to affect Meliá.
18	Takeover Bids for Naturgy and MásMóvil	• Australian fund IFM launches EUR5.000m bid for 22.69% of Naturgy. • Polygon fund asks CNMV to review and alter the bid for MásMóvil. • IFM accepts Spanish government conditions in partial bid for Naturgy.
19	Naturgy: Financial Performance	• Naturgy presents "weak" 2020 results • Naturgy may revise its strategic plan upwards due to gas prices. • Bank of America sees upside potential for Naturgy based on fundamentals.
20	PharmaMar, Grifols: Regulatory Approvals and Market Moves in the Pharmaceutical Sector	• EU court annuls European Commission's refusal to market PharmaMar drug. • Grifols starts issuing EUR2.000m bonds to buy Biotest. • PharmaMar announces approval of lurbinectedin for lung cancer in Australia.
21	Repsol: Financial Performance	• Repsol: Net loss of EUR3.289m in 2020. • Repsol reports a loss of EUR711m in Q4 due to exploration and production provisions • Repsol posts a loss of EUR94m in Q3 due to provisions and lower refining margins.
22	Aena: Financial Performance	• JPMorgan raises Aena's target price to EUR155 from EUR135. • Aena risks a revenue cut of up to EUR2.000m due to rents. • Aena loses EUR170.7m in 1H as passenger traffic plummets due to the pandemic.
23	Enagás, Endesa, Iberdrola, Red Eléctrica: Regulatory and Market Challenges in the Energy Sector	• Spanish electric utilities will remain under pressure in the stock market • Spanish government measures are bad news for the electric sector. • Spain's electricity price closes February with a 52% drop vs January
24	BBVA, CaixaBank, Banco Sabadell: Layoffs and Restructuring	• CaixaBank proposes to unions a redundancy plan affecting 8,291 employees. • Banco Santander closes its redundancy plan with 3,572 voluntary exits and 19 dismissals • Sabadell prepares an adjustment plan affecting 2,000 employees
25	Inditex, Acerinox: Market Performance and Strategic Developments in the Post-Covid Context	• Inditex reopens 94% of its stores worldwide after Covid-19 pandemic. • Sale of Nippon Steel in Acerinox is negative, but expected. • Inditex stock already prices in a strong business recovery.

Table A2: LLM clustering. Sample of 3 articles for each cluster.

#	Title	Articles
0	Demand, Minor, Positive	• Meliá's recovery will be fast, but it will not be completed until 2023 • Tourism sector aid in Spain will have a limited impact on listed companies • Spanish airports will recover pre-pandemic traffic by the end of 2025
1	Demand, Minor, Negative	• Tallgrass will contribute fewer dividends to Enagás -JPMorgan Cazenove • Aena's stock decline is due to sector visibility -Sabadell • ObservaTUR believes Spain's economic situation will worsen and calls for more measures
2	Demand, Major, Positive	• Solaria invests EUR220m in Europe's largest photovoltaic park • Acciona will build Sao Paulo metro line for EUR2.3 billion • Inditex returns to profit in H1 and continues to recover from the pandemic
3	Demand, Major, Negative	• Passenger traffic at Aena airports falls 79.9% year-on-year in September • UPDATE: Naturgy's net profit falls 45.6% in 9m due to Covid-19 impact • Possible capital increase by IAG already priced in
4	Supply, Minor, Positive	• Repsol returns to profit in Q2 due to crude price increase • Naturgy receives LNG supply contract for ships for 2 years in Spain • Acciona Energía starts up 238 MW photovoltaic complex in Chile
5	Supply, Minor, Negative	• Enagás operating results worse than expected -Bankinter • IFM rules out extending acceptance period for Naturgy takeover bid and changing conditions • Changes in Siemens Gamesa's onshore wind business will take time
6	Supply, Major, Positive	• Capital Energy wins renewable auction in Spain • Repsol expects to start exploiting its huge gas reserve in Brazil in 2026 • Repsol will invest EUR657m to expand its industrial complex in Sines, Portugal
7	Supply, Major, Negative	• Iberdrola halts $1.2 billion investment in Mexico • 85% of Acciona workers at Nissan agree to contract termination • CaixaBank reduces workforce adjustment by 500 employees to 7,791 -Source
8	Financial, Minor, Positive	• Norwegian fund Norges Bank takes 1.14% stake in Naturgy amid IFM takeover bid • Sabadell closes green bond issue for EUR500m -Source • CaixaBank-Bankia merger goals are credible -Deutsche Bank
9	Financial, Minor, Negative	• UPDATE2: Bankia's profit falls 57.6% in 2020 due to provisions for pandemic impact • Iberdrola bond spreads will not be affected by Galán's indictment for now • Court maintains precautionary suspension of rent payments to Aena
10	Financial, Major, Positive	• UPDATE: Endesa's net profit soars in 2020 due to lower impairment charges • Telefónica will reduce debt by EUR5bn after closing Virgin Media and O2 merger • Fluidra buys US company S.R. Smith for $240m

11	Financial, Major, Negative	• UPDATE3: Banco Santander reports EUR8.771bn loss in 2020 due to Covid charges • Bankinter downgrades Grifols recommendation to neutral from buy • BBVA reduces layoffs to 3,361 and proposes early retirement at 52 with 65% salary
12	Technology, Minor, Positive	• Siemens Gamesa to supply turbines for 298MW wind farm in the US • Repsol and TÃĺcnicas Reunidas team up to develop decarbonization technologies • European Commission funds Repsol and Enagás renewable hydrogen project
13	Technology, Minor, Negative	• Cellnex and REE apply for EU funds to develop rural mobile networks
14	Technology, Major, Positive	• Telefónica and Allianz partner to deploy fiber in Germany • Iberdrola partners with Cosmo to develop 600 MW of offshore wind in Japan • Telefónica estimates 5G network will require over EUR6bn in Spain
15	Technology, Major, Negative	
16	Policy, Minor, Positive	• Enagás promotes 34 hydrogen and 21 biomethane proposals to recovery funds • Iberdrola president sees need to reform taxation to make renewables competitive • New electricity tariff in Spain aims to change consumer habits -Experts
17	Policy, Minor, Negative	• Spanish government measures hurt Iberdrola -IG • Spanish government plans law to reduce CO2 price impact on electricity bills -Source • Iberdrola CEO criticizes electricity reform in Spain for "unexpected charges"
18	Policy, Major, Positive	• Endesa is Spain's future green leader, but trades at a discount • TCI fund supports ACS's interest in ASPI and will reject Italy's offer • Cellnex acquisition in France reassures investors -Berenberg
19	Policy, Major, Negative	• Sabadell does not expect improvement in partial takeover bid price for Naturgy • Renta 4 downgrades Naturgy to underweight after government measures • Bankinter warns of uncertainties over Iberdrola stock

A.6 Function Calling with Llama-3

Algorithm 4. Function Calling Workflow for Llama-3

Require: $\mathcal{D}$: Dataset of news articles

Ensure: Structured JSON output for each article

1: Initialize Llama-3 model via GroqCloud API

2: **for** each article $i \in \mathcal{D}$ **do** ▷ Iterate over each article in the dataset

3: **Message: System** ▷ Define the role and task for the LLM

"*You are a function calling LLM that analyzes business news in Spanish. For every article, identify the firms that are directly affected by the news and classify the shocks in type, magnitude and direction*"

4: **Message: User** ▷ User provides the article text as input

Content: prompt P_i containing the text of article i

5: **Tool: news_parser** ▷ Define the `news_parser` function

Parameters: {`firms`: `array` of objects}, where each object contains:

- `firm`: `string` ("*each one firm in* ***`firms`***")
- `ticker`: `string` ("*stock market ticker*")
- `shock_type`: `enum` {demand, supply, financial, policy, technology}
- `shock_magnitude`: `enum` {minor, major}
- `shock_direction`: `enum` {positive, negative}

6: Send initial messages and tools to Llama-3 ▷ Initiate interaction with the LLM

7: Retrieve response from Llama-3 ▷ Get the initial response from the LLM

8: **if** Function call is requested by Llama-3 **then** ▷ Check if the LLM needs to call a function

9: Execute `news_parser` function with provided arguments ▷ Run the function

10: Append function response to the conversation ▷ Include function output in the dialogue

11: Send updated messages to Llama-3 ▷ Continue the conversation with new information

12: Retrieve final response from Llama-3 ▷ Get the final output from the LLM

13: **end if**

14: Extract and store structured JSON output ▷ Save the processed data

15: **end for**

A.7 Why not using a different benchmark?

In evaluating our novel Large Language Model (LLM) methodology for classifying news-implied firm-specific shocks, it is imperative to establish a robust and relevant benchmark. Our chosen benchmark involves transforming news articles into high-dimensional vector embeddings followed by clustering these

embeddings using the KMeans algorithm. This section delineates the rationale behind selecting KMeans clustering of vector embeddings over other potential benchmarks such as sentiment analysis and topic modeling.

Why not Sentiment Analysis as a benchmark?

Sentiment analysis is a widely recognized technique in natural language processing that aims to determine the emotional tone conveyed in a text, typically categorizing content as positive, negative, or neutral. While sentiment analysis provides a straightforward approach to gauging the general tone of news articles, it falls short in several critical aspects when juxtaposed with our objectives.

First, sentiment analysis is not sufficiently granular. Our LLM methodology classifies news articles into 20 distinct categories of economic shocks while sentiment analysis classifies articles in a coarse manner, typically into positive, negative, or neutral categories, which is inadequate for benchmarking a detailed classification model.

Second, sentiment analysis predominantly focuses on the linguistic and emotional aspects of the text, which do not necessarily correlate with the economic impact on firms. For instance, a neutral-toned article could describe a significant economic event, while a positive sentiment might not always translate to favorable economic outcomes. Consequently, the sentiment does not provide direct insights into the economic consequences, making it an economically irrelevant benchmark for our purposes.

Third, sentiment analysis algorithms are often sensitive to linguistic subtleties, leading to inconsistent results across different languages and contexts. For example, sarcasm or idiomatic expressions can distort sentiment scores, undermining the reliability of sentiment analysis as a benchmark. This variability poses a challenge for standardization, especially in a multilingual context. For instance, the sentiment derived from analyzing the text in English may significantly differ from the sentiment in Spanish.

Fourth, sentiment analysis is not robust in the sense that different sentiment analysis tools yield divergent assessments of the same text. As shown below, we observe considerable differences in the identified sentiment when applying multiple sentiment analysis providers to a specific article. This lack of consistency undermines the reliability of sentiment analysis as a benchmark, making it unsuitable for our purposes.

> *Sentiment analysis is highly sensitive to the specific tool or model employed. Here, we demonstrate this by analyzing a piece of business news using various popular sentiment analysis tools:* `TextBlob`*,* `text2data`*,* `VADER`*, and* `FinBERT`*. The methods vary significantly in both their approach to sentiment determination and the output they provide, as illustrated below.*[8]

[8]Note that applying Loughran-Macdonald is not recommended in for short texts as it yields sparse results. For example, in the example we are considering, it outputs a category distribution that only loads on "*Strong Modal*", which is not a really useful analysis.

```
LM_Scores = {'Negative': 0, 'Positive': 0, 'Uncertainty': 0, 'Litigious': 0, 'Strong_Modal': 2,
```

Example 3: A news article about Telefónica and Cellnex | Sentiment: `TextBlob`

Cellnex will face more competition in Europe **Score: 0.50**

Telefónica's (TEF.MC) subsidiary, Telxius Telecom, has agreed to sell its telecommunications tower division in Europe and Latin America to American Tower (AMT), which will expand the latter's presence in Europe and increase competition for the Spanish wireless telecommunications group Cellnex Telecom (CLNX.MC), according to Equita Sim. **Score: 0.00** *The transaction "represents the entry of a new independent tower operator into the Spanish market and potentially more competition for future growth in the European market as well," says the brokerage firm.* **Score: 0.06**

Overall **Score: 0.085**

Note: `TextBlob` *is a general-purpose sentiment analysis tool that relies on a pre-built lexicon to assess the polarity of the text. It computes a sentiment score ranging from -1 to 1, where -1 signifies a negative sentiment, 1 indicates a positive sentiment, and 0 represents a neutral sentiment. The methodology behind* `TextBlob` *focuses on tokenizing the input into words and phrases, which are compared against its built-in polarity dictionary.*

Example 4: A news article about Telefónica and Cellnex | Sentiment: `text2data`

Cellnex will face more competition in Europe **Score: 0.145**

Telefónica's (TEF.MC) subsidiary, Telxius Telecom, has agreed to sell its telecommunications tower division in Europe and Latin America to American Tower (AMT), which will expand the latter's presence in Europe and increase competition for the Spanish wireless telecommunications group Cellnex Telecom (CLNX.MC), according to Equita Sim. **Score: -0.512** *The transaction "represents the entry of a new independent tower operator into the Spanish market and potentially more competition for future growth in the European market as well," says the brokerage firm.* **Score: -0.560**

Overall **Score: -0.61**

`'Weak_Modal': 0, 'Constraining': 0, 'Complexity': 0}`

Note: `text2data` *employs scientific deep learning NLP methods to analyze sentiment. Every sentence is split into smaller chunks and represented as a tree structure, capturing the syntactic relationships between words and phrases. To determine the final sentiment score,* `text2data` *uses probabilistic methods based on a pre-trained data model, providing an output score between -1 and 1, where -1 is negative and 1 is positive.*

Example 5: A news article about Telefónica and Cellnex | Sentiment: `VADER`

Cellnex will face more competition in Europe **Score: 0.00**

Telefónica's (TEF.MC) subsidiary, Telxius Telecom, has agreed to sell its telecommunications tower division in Europe and Latin America to American Tower (AMT), which will expand the latter's presence in Europe and increase competition for the Spanish wireless telecommunications group Cellnex Telecom (CLNX.MC), according to Equita Sim. **Score: 0.69** *The transaction "represents the entry of a new independent tower operator into the Spanish market and potentially more competition for future growth in the European market as well," says the brokerage firm.* **Score: 0.57**

Overall **Score: 0.81**

Note: `VADER` *(Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool uses a combination of lexical features (i.e., words) that are generally classified as having positive, negative, or neutral valence.* `VADER` *produces four sentiment metrics: positive, negative, neutral, and a compound score. The compound score is a normalized, weighted composite score that ranges from -1 to 1, indicating the overall sentiment of the text. In this example, we provide the compound measure sentence by sentence and for the whole text.*

Example 6: A news article about Telefónica and Cellnex | Sentiment: `FinBERT`

Cellnex will face more competition in Europe **Negative, 0.75**

Telefónica's (TEF.MC) subsidiary, Telxius Telecom, has agreed to sell its telecommunications tower division in Europe and Latin America to American Tower (AMT), which will expand the latter's presence in Europe and increase competition for the Spanish wireless telecommunications group Cellnex Telecom (CLNX.MC), according to Equita Sim. **Neutral, 0.98** *The transaction "represents the entry of a new independent tower operator into the Spanish market and potentially more competition for future growth in the European market as well," says the brokerage firm.* **Negative, 0.81**

Overall **Negative, 0.94**

Note: `FinBERT` is a domain-specific transformer-based model trained on financial texts. Unlike the previous models, `FinBERT` provides both a sentiment classification (Positive, Negative, Neutral) and a confidence score ranging from 0 to 1, representing the model's certainty about the sentiment classification.

Why not Topic Modeling as a benchmark?

Topic modeling, particularly techniques like Latent Dirichlet Allocation (LDA), decomposes text into a set of latent topics based on word co-occurrences. Topic modelling offer a more granular approach compared to sentiment analysis and could potentially offer a valid benchmark for our purpose. However, we argue that transforming news articles into vector embeddings and subsequently clustering them using KMeans offers a more balanced approach than topic modeling.

Topic models rely on bag-of-words representations, which disregard the order and context of words. This limitation hampers the model's ability to capture complex semantic relationships and contextual nuances essential for accurately identifying economic shocks. Consequently, topic models may overlook subtle but economically significant information present in the text. On the other hand, vector embeddings encapsulate rich semantic information by capturing the relationships between words in a continuous vector space. Unlike topic models, which are confined to word co-occurrences, embedding models, particularly transformer-based, generate context-dependent representations, allowing for a nuanced understanding of polysemy and context. This means that the same word can have different embeddings depending on the context of the sentence, such as "*Apple*" in "*Apple is a leading tech company*" versus "*Apple is a type of fruit*".

An important advantage of vector embeddings is that they scale efficiently with large corpora and can be generated at various granularities, including word, sentence, or document levels. This scalability makes embeddings highly adaptable for diverse downstream tasks such as clustering, classification, and similarity detection. In contrast, topic models often require extensive manual tuning and become computationally expensive with larger datasets, limiting their practicality for extensive analyses. This makes embeddings a superior choice for grouping news articles and analyzing their economic implications, as compared to the relatively rigid and broad classifications produced by topic models.

It is true, however, that topic models excel at grouping articles based on shared themes, offering a straightforward way to identify and interpret these themes by examining the common content of the grouped articles. This interpretability is a key advantage of topic models, as it allows for clear labeling of themes. In contrast, vector embeddings lack inherent interpretability at the dimension level. The individual dimensions of an embedding do not have an intuitive meaning, making it challenging to directly understand the relationships they capture. However, this limitation can be mitigated by clustering the embeddings to then apply a similar interpretive process as with topic models: analyzing the articles within each cluster to infer the common patterns. As demonstrated in our analysis, these clusters often correspond to firm-specific or industry-specific topics, offering valuable insights into economic relationships and forming a valuable benchmark for our LLM's classification of firm-specific shocks.

Lastly, using embeddings as a benchmark is particularly compelling because they represent the foundational layer of an LLM. The first step an LLM's processing pipeline is to transform the text that it is fed into high-dimensional embeddings for further processing. By benchmarking against embeddings, we ensure a direct and relevant comparison between the foundational representations used by LLMs and our specialized classification methodology. This comparison highlights the added value of the LLM's capacity to convert these semantic representations (i.e: the vector embeddings) into economically meaningful classifications. (i.e: our news-implied firm-specific shock classifications).

In summary, KMeans clustering of vector embeddings offers a robust and economically relevant benchmark for our LLM-based methodology. It provides a rich semantic representation, context-dependent flexibility, and scalability that surpass sentiment analysis and topic modeling. Additionally, its alignment with the underlying architecture of LLMs ensures a meaningful comparison. As demonstrated in our analysis, the clusters derived through this approach are predominantly firm or industry-specific, thereby offering a suitable and superior benchmark against which to measure the effectiveness of our granular classification of news-implied firm-specific shocks.

A.8 Trading Intensity

The extraordinary performance of our proposed LLM-based methodology warrants a careful examination of its implementation costs and practical viability. While our primary objective has been to develop a framework that better captures the economic content of news articles and their subsequent market

impact, the practical implementation of such strategies necessarily involves trading frictions that could affect their real-world efficacy. In this section, we analyze the trading intensity patterns of both methodologies to provide a more complete assessment of their relative merits and to understand how transaction costs might influence their comparative advantages. We begin by examining the temporal evolution of open positions for both approaches, which provides insights into their underlying trading dynamics and stability characteristics. This analysis is followed by detailed trading intensity metrics and concludes with a reassessment of portfolio statistics after accounting for transaction costs.

[Insert Figure A6 about here]

The temporal evolution of open positions, as illustrated in Figure A6, reveals fundamental differences in the stability and reliability of trading signals generated by KMeans versus LLM-based clustering approaches. The KMeans implementation exhibits pronounced volatility in position management, particularly evident in the Greedy algorithm's behavior, which shows extreme fluctuations ranging from 6 to 105 positions. This erratic pattern suggests that KMeans-detected clusters are highly sensitive to market noise and potentially capture transient correlations rather than fundamental relationships. The substantial divergence between Greedy and Stable algorithms under KMeans further underscores the method's instability, as even minor variations in cluster selection criteria lead to dramatically different trading decisions. In stark contrast, the LLM-based approach demonstrates remarkably more coherent and stable position management. Both Greedy and Stable algorithms maintain more closely aligned position counts, typically ranging between 20 and 75 positions, with highly correlated temporal movements. This convergence in behavior between algorithms suggests that LLM-identified clusters capture more fundamental and persistent market relationships. Particularly telling is the test period performance, where KMeans exhibits increased position volatility and extreme spikes, while the LLM approach maintains consistent position patterns across both algorithms. This stability in the out-of-sample period provides strong evidence that LLM-derived signals, grounded in economic analysis of firm-specific shocks, generalize more effectively to unseen data.

[Insert Table A3 about here]

The trading intensity metrics, detailed in Table A3, provide quantitative validation of the structural differences between KMeans and LLM clustering approaches. Under KMeans, the dramatic disparity between Greedy and Stable algorithms (averaging 40.1 versus 10.77 positions, with standard deviations of 18.59 and 6.41 respectively) reflects the method's fundamental instability. More concerning is the Stable algorithm's exceptionally high Changes/Position ratio (3.228 versus 0.798 for Greedy), indicating frequent position adjustments necessitated by the transient nature of KMeans-identified clusters. The LLM implementation demonstrates substantially more balanced and stable metrics across both algorithms. Average position counts converge (31.8 for Greedy, 26.61 for Stable) with more moderate standard deviations

(14.84 and 12.16), suggesting that both aggressive and conservative cluster selection approaches identify similar, fundamentally-driven trading opportunities. The more balanced Changes/Position ratios (1.234 and 1.473) and consistent turnover rates (approximately 39% for both algorithms) indicate that LLM-identified clusters require less frequent rebalancing, supporting the hypothesis that they capture more persistent market relationships.

[Insert Table A4 about here]

Finally, the introduction of trading costs impacts the performance metrics of both clustering approaches (see Table A4), though with notably different implications for their practical viability. The KMeans-based strategy exhibits visible performance degradation, particularly evident in the test period where both algorithms generate losses (Greedy: -4.1%, Stable: -6.8% average annual returns). This deterioration is accompanied by elevated risk metrics, with the Stable algorithm showing particularly concerning characteristics including high standard deviation (14.2%) and extreme kurtosis (14.74) in the test period, suggesting frequent occurrence of extreme returns. In contrast, the LLM-based approach demonstrates superior resilience to trading costs, maintaining more stable performance characteristics across all periods. Most notably, in the test period, the strategy maintains its positive performance (Greedy: 19.0%, Stable: 24.7% annual returns) with substantially lower risk metrics (standard deviations of 6.2% and 7.0% respectively). The LLM approach's more moderate VaR and CVaR measures compared to KMeans further underscore its superior risk management characteristics under transaction costs. This stark contrast in net performance can be attributed to the fundamentally different nature of the signals generated by each approach. While KMeans' statistically-driven clusters require frequent rebalancing that amplifies transaction costs, the LLM's economically-motivated clusters appear to identify more persistent price patterns that remain profitable even after accounting for trading frictions. However, it is worth noting that neither approach was explicitly optimized for transaction cost efficiency, suggesting potential for further improvement through cost-aware portfolio construction. These results highlight that while our LLM-based news parser successfully captures predictable market reactions to news articles, practitioners implementing such strategies would benefit from incorporating transaction costs into their optimization framework.

TABLE A3: Trading Intensity Analysis: Model Comparison

(A) Panel A: KMeans

Split	Algorithm	# Open Positions				Trading Activity (%)		Trading Costs (%)	
		Avg.	Std.	Max	Min	Turnover	Changes/Pos.	Cost	Active
All	*Greedy*	40.1	18.59	105	6	32.03	0.798	0.0320	100.0
	Stable	10.77	6.41	30	0	34.75	3.228	0.0347	99.1
Train	*Greedy*	36.4	19.33	88	7	30.59	0.840	0.0306	100.0
	Stable	9.89	5.93	27	0	33.73	3.412	0.0337	98.2
Validation	*Greedy*	48.4	10.00	80	30	31.39	0.649	0.0314	100.0
	Stable	12.34	6.05	30	1	33.42	2.708	0.0334	100.0
Test	*Greedy*	38.8	21.74	105	6	35.86	0.925	0.0359	100.0
	Stable	10.84	7.47	28	1	39.30	3.626	0.0393	100.0

(B) Panel B: LLM

Split	Algorithm	# Open Positions				Trading Activity (%)		Trading Costs (%)	
		Avg.	Std.	Max	Min	Turnover	Changes/Pos.	Cost	Active
All	*Greedy*	31.8	14.84	75	4	39.21	1.234	0.0392	100.0
	Stable	26.61	12.16	56	3	39.18	1.473	0.0392	100.0
Train	*Greedy*	29.9	16.34	75	4	40.42	1.351	0.0404	100.0
	Stable	25.54	12.90	56	3	40.45	1.584	0.0404	100.0
Validation	*Greedy*	37.0	7.69	58	24	38.43	1.039	0.0384	100.0
	Stable	31.38	6.82	50	17	37.95	1.209	0.0379	100.0
Test	*Greedy*	29.7	16.24	75	6	37.56	1.264	0.0376	100.0
	Stable	23.43	13.71	54	3	37.85	1.615	0.0378	100.0

Note: This table presents trading intensity metrics for both Greedy and Stable algorithms across different data splits for two different models: KMeans (Panel A) and LLM (Panel B). The metrics are computed at a daily frequency. The '# Open Positions' columns report position-related statistics: 'Avg.' shows the mean number of concurrent open positions per day, 'Std.' represents their standard deviation, while 'Max' and 'Min' indicate the maximum and minimum number of positions held simultaneously. Under 'Trading Activity (%)', 'Turnover' is calculated as the sum of absolute changes in position sizes divided by the total portfolio size, expressed as a percentage; formally, $Turnover_t = 100 \times (\sum_i |w_{i,t} - w_{i,t-1}|)/(\sum_i |w_{i,t}|)$*, where* $w_{i,t}$ *represents the position size in asset* i *at time* t*. 'Changes/Pos.' represents the average number of modifications per position per day, computed as the daily turnover divided by the average number of positions, providing insight into how actively individual positions are managed. The 'Trading Costs (%)' section reports 'Cost' as the average daily implementation shortfall (computed as the product of daily turnover and a transaction cost parameter of 10 basis points) expressed in percentage terms, while 'Active' shows the percentage of trading days with at least one open position. All metrics are first computed daily and then averaged over their respective periods, except for Max and Min positions which represent the absolute extremes over each period.*

TABLE A4: Portfolio Statistics Comparison: KMeans vs LLM Clustering (net of Trading Costs)

(A) Panel A: Statistics of $\mathcal{P}_{\text{KMeans}}$

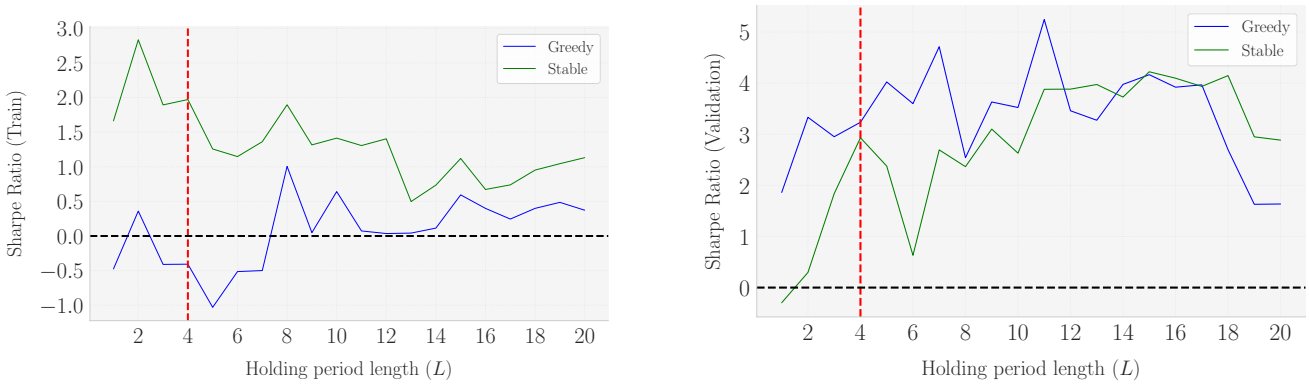
Split	Algo.	Cum. Ret.	Avg. Ret.	St. Dev.	Sharpe Ratio	Sortino Ratio	Max. DD	Calmar Ratio	Skew.	Exc. Kurt.	VaR 95%	CVaR 95%
All	*Greedy*	0.963	-2.9	9.6	-0.3	-0.3	-9.5	-0.3	-0.46	4.00	-13.7	-23.4
	Stable	1.329	24.4	16.8	1.3	1.5	-8.3	2.9	0.18	5.08	-23.0	-36.8
Train	*Greedy*	0.911	-13.2	11.6	-1.2	-1.1	-9.5	-1.4	-0.52	2.72	-18.7	-28.9
	Stable	1.182	28.9	19.7	1.3	1.4	-8.3	3.5	-0.23	3.24	-30.6	-44.0
Validation	*Greedy*	1.058	17.1	7.3	2.2	2.2	-4.0	4.3	-0.48	1.10	-10.2	-16.6
	Stable	1.115	35.7	13.3	2.3	2.6	-4.2	8.6	-0.23	1.85	-19.3	-29.0
Test	*Greedy*	0.988	-4.1	6.8	-0.6	-0.8	-5.3	-0.8	1.76	5.10	-8.2	-10.4
	Stable	0.979	-6.8	14.2	-0.5	-0.6	-5.6	-1.2	2.49	14.74	-19.4	-27.4

(B) Panel B: Statistics of $\mathcal{P}_{\text{LLM}}$

Split	Algo.	Cum. Ret.	Avg. Ret.	St. Dev.	Sharpe Ratio	Sortino Ratio	Max. DD	Calmar Ratio	Skew.	Exc. Kurt.	VaR 95%	CVaR 95%
All	*Greedy*	1.152	11.5	9.6	1.1	1.4	-7.6	1.5	1.47	9.93	-14.4	-19.6
	Stable	1.200	15.0	8.6	1.6	1.9	-7.2	2.1	0.29	2.23	-12.1	-17.4
Train	*Greedy*	1.040	6.2	11.4	0.5	0.7	-7.6	0.8	1.65	8.97	-16.2	-21.6
	Stable	1.101	15.9	9.9	1.5	1.7	-7.2	2.2	0.18	1.68	-14.1	-20.3
Validation	*Greedy*	1.054	16.2	8.2	1.8	2.3	-3.3	4.9	0.16	1.31	-10.9	-17.2
	Stable	1.013	3.8	7.0	0.5	0.6	-2.2	1.7	0.22	1.31	-11.7	-15.3
Test	*Greedy*	1.054	19.0	6.2	2.8	3.5	-1.6	11.9	1.35	7.85	-7.5	-10.6
	Stable	1.069	24.7	7.0	3.1	4.7	-1.3	18.6	0.86	1.99	-10.1	-11.6

Note: Portfolio statistics of trading strategies based on clusters obtained from KMeans (Panel A) and LLM (Panel B) approaches. The statistics provided include performance metrics (Cumulative Return, Average Return (%)), risk measures (Standard Deviation (%), Maximum Drawdown (%), Value at Risk (%), Conditional Value at Risk (%)), risk-adjusted performance ratios (Sharpe Ratio, Sortino Ratio, Calmar Ratio), and return distribution characteristics (Skewness, Excess Kurtosis). These statistics are provided for both cluster-selection algorithms: Greedy and Stable. Except for the Cumulative Return, all returns are annualized. The Sharpe Ratio is computed using the daily returns, assuming 252 trading days in a year. The Sortino Ratio is calculated using the daily downside returns. The Maximum Drawdown is the maximum loss from a peak to a trough. The Calmar Ratio is the ratio of the annualized return to the maximum drawdown. Skewness measures the asymmetry of the return distribution, while Kurtosis quantifies the tails' thickness. The Value at Risk (VaR) and Conditional Value at Risk (CVaR) are calculated at a 95% confidence level. All returns are calculated net of transaction costs. We implement a transaction cost estimate of 10 basis points per trade,. The Greedy algorithm longs (shorts) clusters that maximize (minimize) the cluster-average-SR in the validation sample subject to a positivity (negativity) constraint, while the Stable algorithm longs (shorts) clusters that minimize the rank difference between the training and validation rankings of the cluster-average-SR's subject to a positivity (negativity) constraint, which is now imposed on both sample splits. In both algorithms, the cardinality of each leg is upper-bounded by a hyperparameter θ. The holding period of the beta-neutral positions is set to $L = 4$ trading days for both approaches. The number of traded clusters is $\theta = 0.5k = 13$ for KMeans ($k^* = 26$ clusters) and $\theta = 0.5k = 10$ for LLM ($k^* = 20$ clusters). The selection criteria for these hyperparameters (L, θ) is based on maximizing the Sharpe Ratios of the train and validation samples.

FIGURE A1: Sharpe Ratios in the train and validation splits as a function of L (KMeans)

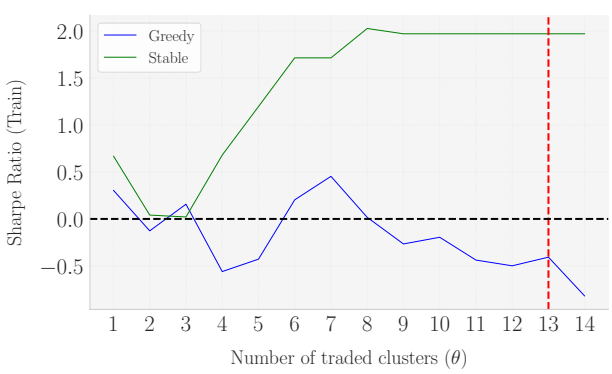


(A) Plot of $SR^{\mathcal{P}^{tr}}(L)$ over a grid of L

(B) Plot of $SR^{\mathcal{P}^{val}}(L)$ over a grid of L

Note: This figure shows the Sharpe Ratios (SR) as a function of the holding period length (L) for the KMeans clustering method in the training (Panel a) and validation (Panel b) splits. In Panel (a), the Sharpe Ratios in the training set indicate that lower values of L (less than 4) maximize performance. Conversely, in Panel (b), the validation set shows higher Sharpe Ratios for longer holding periods. The choice of $L = 4$ represents a balanced compromise, providing a stable Sharpe Ratio profile across both splits, ensuring consistent in-sample performance without introducing lookahead bias.

FIGURE A2: Sharpe Ratios in the train and validation splits as a function of θ (KMeans)

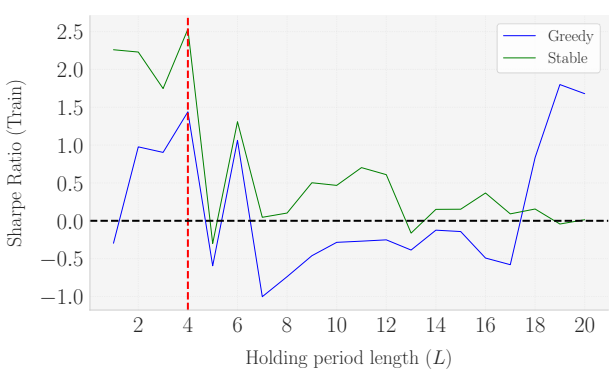


(A) Plot of $SR^{\mathcal{P}^{tr}}(\theta)$ over a grid of θ

(B) Plot of $SR^{\mathcal{P}^{val}}(\theta)$ over a grid of θ

Note: This figure illustrates the Sharpe Ratios (SR) as a function of θ, the upper bound on the number of traded clusters, for the KMeans clustering method in the training (Panel a) and validation (Panel b) splits. In Panel (a), the Sharpe Ratios in the training set show a trend of increasing stability and maximizing performance as θ approaches its upper limit. Similarly, Panel (b) displays a consistent pattern in the validation set, where higher values of θ lead to convergence at the highest and most stable Sharpe Ratios. The choice of $\theta = 13$ (i.e: $\lfloor 0.5 \cdot 26 \rfloor$) reflects this observed stability and optimization, providing a balanced and robust selection for the portfolio strategy.

FIGURE A3: Sharpe Ratios in the train and validation splits as a function of hyperparameters (LLM)

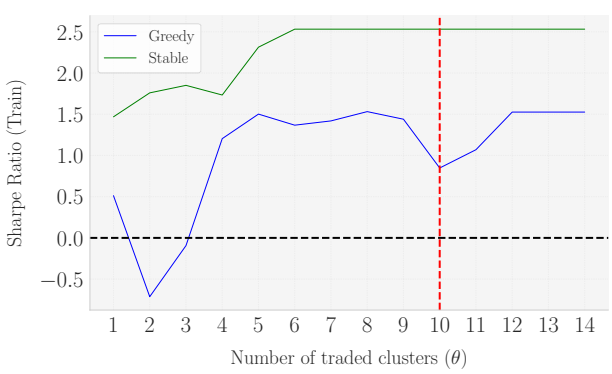


(A) Plot of $SR^{\mathcal{P}^{tr}}(L)$ over a grid of L

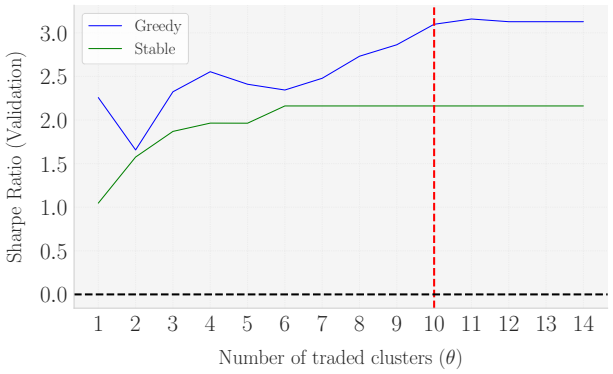
(B) Plot of $SR^{\mathcal{P}^{val}}(L)$ over a grid of L

Note: This figure shows the Sharpe Ratios (SR) as a function of the holding period length (L) for the LLM clustering method, across the training (Panel a) and validation (Panel b) splits. In Panel (a), the Sharpe Ratios in the training set reach their maximum at $L = 4$, suggesting shorter holding periods are more effective for maximizing performance. Conversely, Panel (b) illustrates that longer holding periods yield higher Sharpe Ratios in the validation set. The choice of $L = 4$ serves as a compromise, balancing the trade-off between maximizing SR in both splits and providing a stable and consistent holding period length for the strategy.

FIGURE A4: Sharpe Ratios in the train and validation splits as a function of θ (LLM)



(A) Plot of $SR^{\mathcal{P}^{tr}}(\theta)$ over a grid of θ



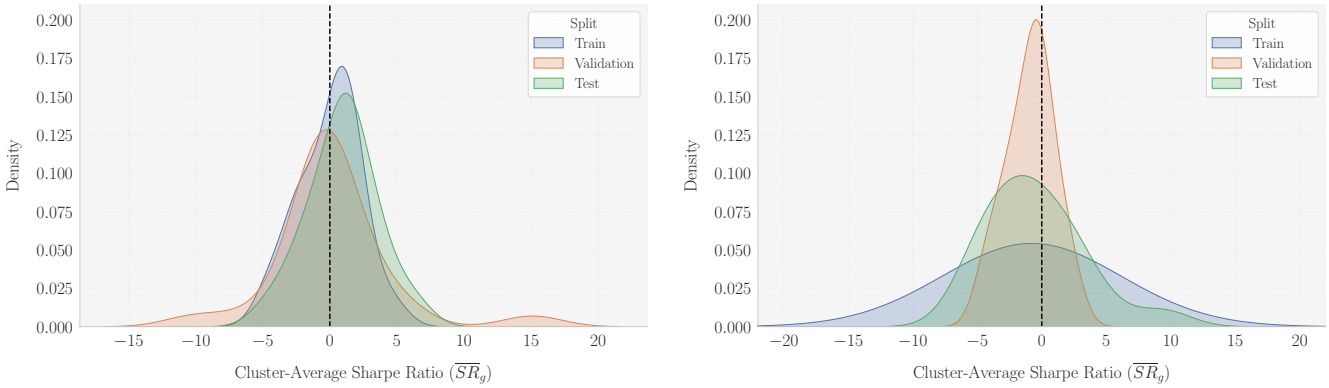
(B) Plot of $SR^{\mathcal{P}^{val}}(\theta)$ over a grid of θ

Note: This figure illustrates the Sharpe Ratios (SR) as a function of θ, the upper bound on the number of traded clusters, for the LLM clustering method in the training (Panel a) and validation (Panel b) splits. In Panel (a), the Sharpe Ratios for the training set indicate a temporary dip at $\theta = 10$ for the Greedy algorithm, yet this value still provides a relatively stable outcome. In contrast, Panel (b) shows that $\theta = 10$ leads to a noticeable increase in Sharpe Ratios for the validation set, particularly benefiting the Greedy algorithm. The choice of $\theta = \lfloor 0.5k \rfloor = 10$ strikes a balance, confirming it as an effective hyperparameter selection for achieving stability in both the training and validation splits with LLM clustering.

FIGURE A5: Distribution of Cluster-Average Sharpe Ratios ($\overline{SR}_g$) by Split

(A) Panel A: KMeans Clustering

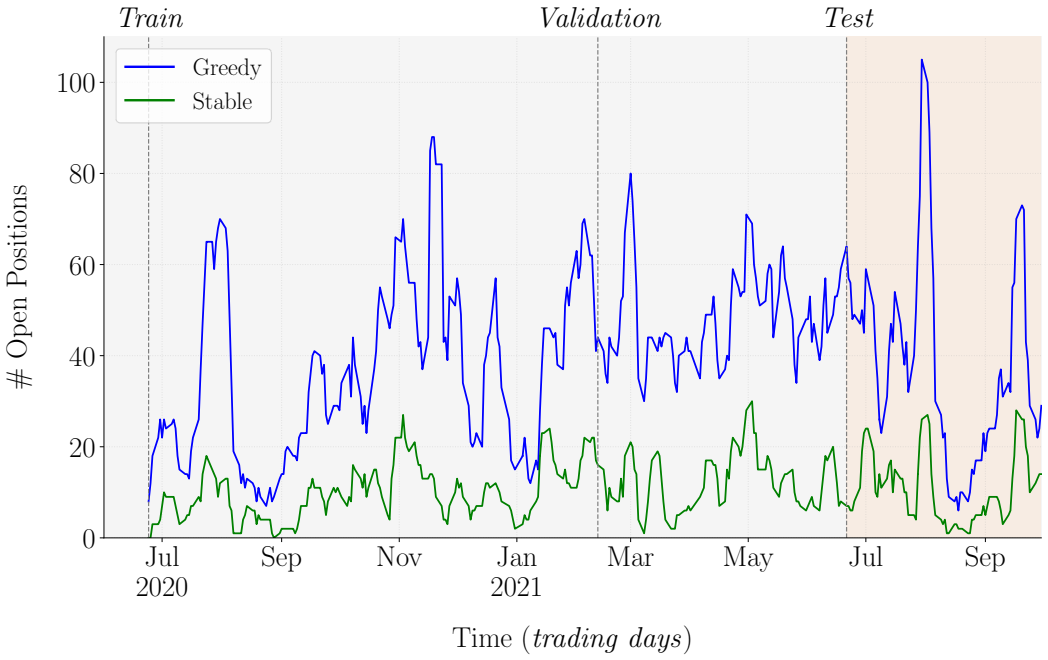
(B) Panel B: LLM Clustering



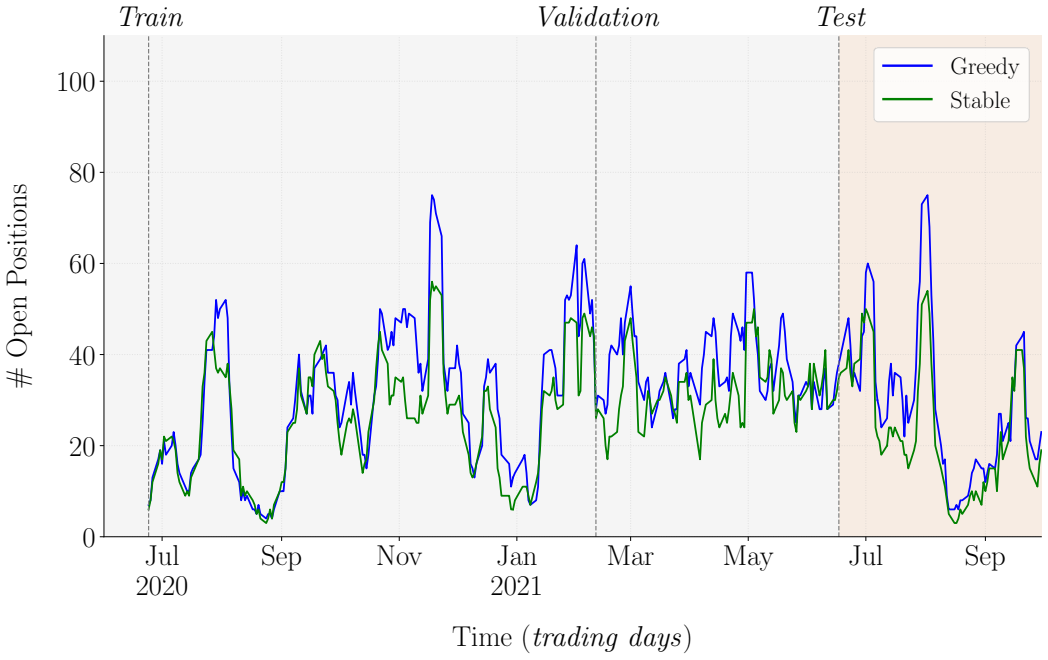
Note: This figure presents the distribution of cluster-average Sharpe Ratios ($\overline{SR}_g$) across training, validation, and test data splits for both KMeans clustering (Panel A) and LLM clustering (Panel B). Each Sharpe Ratio is computed as the average of beta-neutral positions associated with articles in a given cluster. The KMeans approach (Panel A) shows distributions centered around 0 in the validation set, with some outliers exhibiting unusually high or low Sharpe Ratios. The training and test set distributions are slightly right-skewed, suggesting better performance in certain clusters, with no significant outliers. In contrast, the LLM clustering (Panel B) exhibits left-skewed distributions across all splits, indicating a higher frequency of lower Sharpe Ratios. The training data shows fat tails, suggesting extreme values, while the validation data has lighter tails. The test data distribution is more bell-shaped, with Sharpe Ratios concentrated between 5 and 15, indicating stronger performance in some clusters.

FIGURE A6: Evolution of Open Positions: KMeans vs LLM Clustering

(A) Panel A: KMeans Clustering



(B) Panel B: LLM Clustering



Note: This figure shows the daily evolution of the number of open positions for both Greedy (blue) and Stable (green) algorithms across different data splits (Train, Validation, Test) using KMeans clustering (Panel A) and LLM clustering (Panel B). The time period spans from July 2020 to September 2021. Vertical dashed lines separate the different data splits. The Greedy algorithm selects clusters that maximize (minimize) the cluster-average-SR for long (short) positions, while the Stable algorithm minimizes the rank difference between training and validation rankings. The number of traded clusters is $\theta = 0.5k = 13$ *for KMeans (*$k^* = 26$ *clusters) and* $\theta = 0.5k = 10$ *for LLM (*$k^* = 20$ *clusters).*