CONTRASIM: CONTRASTIVE SIMILARITY SPACE LEARNING FOR FINANCIAL MARKET PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present *Contrastive Similarity Space* — **ContraSim** , a novel paradigm designed to establish a global semantic understanding of how daily financial headlines influence market movements. ContraSim works in two steps: **(i) Weighted Headline Augmentation**, which generates augmented financial headlines with known semantic distances from the original, and (ii) **Weighted Self-Supervised Contrastive Learning** (WSSCL), an extension of classical binary contrastive learning that leverages these distances to create a finely tuned embedding space where semantically similar headlines are clustered together. To evaluate ContraSim, we introduce a novel information density metric, **g-kNN**, which measures its ability to inherently group news headlines associated with homogeneous market movement directions. We empirically demonstrate that incorporating ContraSim features into financial forecasting tasks yields a 7% improvement in classification accuracy. Furthermore, we show that ContraSim enables the identification of historical news-days that most resemble the financial headlines of the current day, offering analysts actionable insights to better predict market trends and movements by juxtaposing closest historical occurrences.

1 INTRODUCTION

With recent explosion in the capabilities of Large Language Models (LLMs), researchers have been able to dramatically increase the ability to break down the semantic richness in textual data to be used in downstream tasks. Mature fields such as Sentiment Analysis Devlin et al. [2019], Spam Detection Aggarwal et al. [2022], Machine Translation Vaswani et al. [2017], and many more Liu et al. [2019], Brown et al. [2020], Radford et al. [2019] have been completely revolutionized by the advent of deep LLMs. Likewise, because a key source of information in the the domain of financial market movement prediction is encoded in textual representations (news, reports, social media, etc.), a predictable field of study has been how LLMs can be used to better predict market movement.

It is known that the direction of a stock's price is impacted by a plethora of temporally linked features, like overall market movement, industry trends and company-specific news. It has been a daunting task for researchers to build machine learning algorithms that are able to interpret the complex and noisy feature space of textual financial news, to repeatedly perform well in market movement prediction. Previous models created the majority of their predictive powers by solely looking at historic financial indicators Fischer & Krauss [2018], Sezer & Ozbayoglu [2018]. However, with LLM's ability to create dense feature representations from human text, composite models that utilize financial indicators in conjunction with news, and social-media posts were able to improve predictive performance Saqur [2024], Liu et al. [2021]. Multiple projects have found success doing this by using a mixture of classical and deep learning approaches Ding et al. [2015], Fischer & Krauss [2018], Hu et al. [2018], Sezer & Ozbayoglu [2018], Xu et al. [2018], Liu et al. [2021]. State of the art approaches to stock market prediction is outlined in section 2.

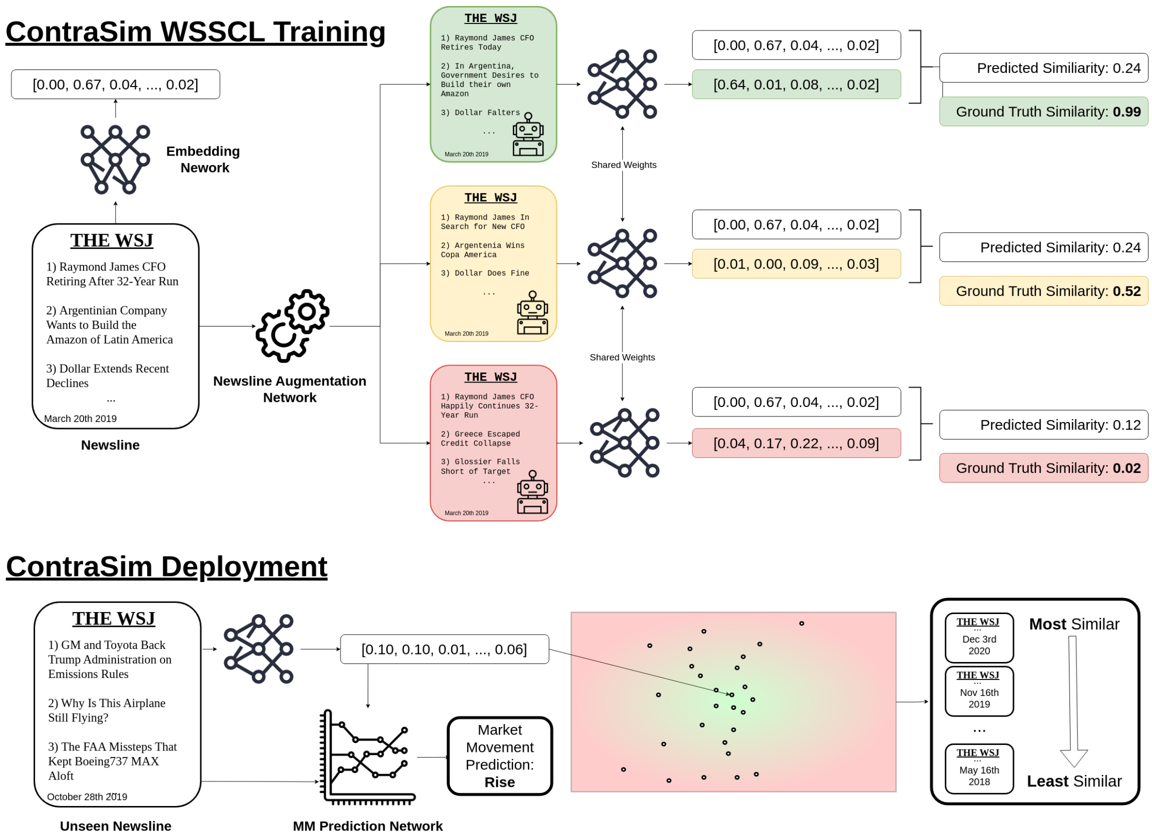


Figure 1: Overview of our proposed Contrastive Similarity (ContraSim) embedding approach. **In training**, we use a LLaMA chat model to generate augmented financial news headlines with varying degrees of semantic similarity to the original. We then use a Weighted Self-Supervised Contrastive Learning (WSSCL) approach to create an embedding space that clusters semantically similar prompts closer together. **In deployment**, the embeddings from the similarity space, can be used to i) Make better predictions on the direction of today's stock movement, ii) Find the most similar financial news to today's.

While composite models that blend financial indicators with language features have improved market movement predictions, they often function as "black boxes." They predict market changes without offering any insight into why a particular prediction was made, making them less useful for financial analysts seeking interpretability. To address this, we propose a Contrastive Self-Supervised Learning approach that not only enhances market movement predictions using financial text data but also preserves interpretability. Our method aims to: a) predict the current day's market direction using Wall Street Journal (WSJ) headlines The Wall Street Journal [2024], and b) provide a ranked list of similar past financial news events.

The idea behind our approach is straightforward. We treat a day's news as a combined list of all WSJ (and other relevant, reputable sources) headlines for that day. For example, a headline like "*Canadian Crude Prices Hit by Keystone Pipeline Shutdown*" (2019-11-05) serves as input, much like other models. However, in addition to predicting market changes, our approach also identifies other days when similar events occurred. For instance, the most similar past headline might be "*Russian Pipeline Shutdown Shifts Balance in Oil Market*" (2019-05-22). This method offers a balance of interpretability and simplicity, allowing analysts to identify patterns in current news and historical contexts without relying on a complex "Explainable AI" (XAI) component.

We propose ContraSim, a method that leverages a novel textual augmentation algorithm powered by LLMs to generate diverse news headlines with varying degrees of semantic similarity to the original. Augmented headlines are assigned similarity scores ranging from 1.0 (high semantic alignment) to 0.0 (completely disjoint meaning). Using these augmented pairs, we introduce Weighted Self-Supervised Contrastive Learning (WSSCL) to build an embedding space where semantically similar headlines are naturally clustered. This embedding algorithm enables the calculation of similarity scores between any two real-world headlines based on their semantic proximity.

This approach is validated through two key findings: a) WSSCL inherently groups headlines associated with similar market directions closer in the embedding space. Even without explicit market movement labels, the model intuitively captures the relationship between headlines and market behavior using an information-gain framework, and b) A large language model (LLM) trained with WSSCL-enhanced embeddings outperforms an LLM relying solely on raw financial headlines for market movement prediction, demonstrating the added value of this semantic embedding strategy.

Contributions : We introduce the *Contrastive Similarity Space Embedding Algorithm* (ContraSim), a method that generates prompt augmentations with meaningful and nuanced similarity coefficients. We demonstrate that:

a) ContraSim enables inter-day financial comparisons, allowing forecasters to identify historic market days similar to the current day.
b) ContraSim learns a mapping between news headlines and market direction in an unsupervised manner. This is evidenced by emergent structures in the embedding space that increase global insight into stock movement – i.e., by identifying similar prompts, we gain insight into why stocks move.
c) The similarity embedding spaces created by ContraSim enhance the performance of financial forecasting classification algorithms when used together.

Organization: Section §2 provides background using related works. Section §3 outlines our main methodologies. Section §4 explicates our experiments and empirical results, alongside an outline for future work and training details. Other additional information like headline transformation A, g-KNN C, and our datasets D, are relegated to the appendix.

2 Related Works

Machine Learning in Financial Forecasting Early approaches to predicting stock market movements relied heavily on classical statistical models. One foundational method, the Autoregressive Integrated Moving Average (ARIMA) Box & Jenkins [1970], utilized time series data to forecast trends. Subsequent models, such as *Generalized Autoregressive Conditional Heteroskedasticity* (GARCH) Bollerslev [1986], *Vector Autoregression* (VAR) Sims [1980], and *Holt-Winters exponential smoothing* Holt [1957], extended these capabilities by capturing more intricate patterns in financial time series. Other notable contributions include techniques for cointegration analysis Engle & Granger [1987], Kalman filtering Kalman [1960], and Hamilton's regime-switching models Hamilton [1989].

While effective, these classical models were primarily limited to tabular datasets and struggled with nonlinear relationships and multimodal inputs. The rise of Large Language Models (LLMs) transformed financial forecasting by enabling the incorporation of richer, more complex data sources. For example, integrating financial news articles Yang et al. [2020], sentiment analysis Yang et al. [2020], social media activity Bollen et al. [2011], and earnings call transcripts Tsai & Wang [2016] significantly enhanced market movement predictions, demonstrating the versatility and power of LLMs in handling diverse financial modalities.

Contrastive Learning Contrastive learning has emerged as a powerful paradigm in unsupervised and self-supervised learning, focusing on representation learning through comparisons. The core idea is to bring similar data points closer in the representation space while distancing dissimilar ones. A key milestone in this field was SimCLR Chen et al. [2020], which used data augmentations and contrastive loss to learn high-quality representations without requiring labels. MoCo He et al. [2020] further advanced this approach by introducing a memory bank to efficiently manage negative examples, making it more scalable for larger datasets. Recent innovations like SimSiam Chen & He [2021] have shown that competitive representations can be learned without relying on negative pairs, streamlining computation and improving accessibility. These advancements are particularly relevant for financial applications, where large-scale and heterogeneous datasets are common, enabling contrastive learning to uncover nuanced relationships in financial data.

3 METHODS

In this section, we introduce ContraSim, a self-supervised contrastive learning algorithm that creates augmented news headlines with fine-grained degrees of similarity to the base. Then using a weighted self-supervised learning paradigm, we create an embedding space, where semantically similar news headlines are clustered together. Additionally, we outline how we can measure the efficacy of ContraSim by using an information density approach in our similarity space to see if there is inherent market-movement knowledge being learned by optimizing for news headline similarity.

3.1 CONTRASIM: CONTRASTIVE SIMILARITY SPACE EMBEDDING ALGORITHM

Here, we formulate the news headline augmentation pipeline and the Weighted Self-Supervised Contrastive Learning (WSSCL) approach that in tandem generate the ContraSim. The contrastive similarity space, generated from ContraSim, is optimized to put the headlines with semantically similar news into local proximity.

We define the news headline dataset as:

$$\mathcal{D}_{\text{news headlines}} = \{(d_i, \mathcal{N}_i) \mid i = 1, 2, \ldots, n\}, \quad \text{where } \mathcal{N}_i = (h_{i1}, h_{i2}, \ldots, h_{im}),\ 10 \leq m \leq 30 \tag{1}$$

Where, n is the total number of news headlines within the news headline dataset, $\mathcal{N}_i$ is news headline object containing a tuple of 10-30 headlines strings h, and d_i is the corresponding date identifier string for a day i. In this context, a news headline is collection of WSJ The Wall Street Journal [2024] (or other relevant, reputable market sources) headlines, however in this paper we explore how well ContraSim performs on other textual domains (e.g. list of movie reviews).

1. Defining the Augmentation Objective Below, we propose a stochastic transformation $T : \mathcal{N} \rightarrow (s, \hat{\mathcal{N}})$, where $\mathcal{N}$ is an input news headline, $\hat{\mathcal{N}}$ is the augmented news headline, and $s \in [0.0, 1.0]$ represents the similarity score between $\mathcal{N}$ and $\hat{\mathcal{N}}$. In subsection 3. we further discuss our implementation details and our process of measuring inter-news headline semantic similarity.

The dominant strategy for creating contrastive embedding spaces defines inter-object relationships in binary terms: two objects are either within the same class or outside the same class. However, for this objective, we do not have access to binary class labels between news headlines, as the similarity between news headlines is inherently regressive and varies along a continuous spectrum. Weighted contrastive approaches, such as Xi et al. [2022], better align with this setting by leveraging nuanced similarity scores to guide the embedding space construction, enabling more accurate representation of the semantic relationships between augmented news headlines.

2. Generating Augmented News Headlines Augmented news headlines are generated through the following discrete actions: i) Rewording the original headline (**Re**), ii) Generating a semantically shifted version (**S**), iii) Negating the original headline (**N**), and iv) Selecting a random headline from a different day (**Ra**).

To achieve these transformations, we leveraged the LLaMA-3-7b-chat model AI [2024], prompting it with carefully crafted instructions tailored to each specific action. For rewording (**Re**), the model was prompted to retain the original meaning of the headline while rephrasing it with alternative wording and sentence structure. For semantic-shifting (**S**), the prompt instructed the model to subtly alter the meaning of the headline, introducing slight semantic deviations while maintaining topical relevance. For negation (**N**), the model was guided to generate a headline that conveyed the direct opposite meaning of the original. By using these tailored prompts, the LLaMA model provided high-quality augmented news headlines that covered a broad spectrum of semantic variations. A further exploration on the specifics of the three (steps (i)-(iii)) headline transformations are expanded upon in appendix section A. Table 1 depict a pedagogical example illustrating these transformations:

Transformation Action	**Example Headline**
Original	*Johnson & Johnson to Buy Surgical Robotics Maker Auris*
Reworded (Re)	*Auris Acquired by Pharmaceutical Giant Johnson & Johnson*
Semantically-Shifted (S)	*Abbott Laboratories Acquires Medical Imaging Specialist Siemens Healthineers*
Negated (N)	*Auris to Sell Off Stake in Surgical Robotics Business to Johnson & Johnson*

Table 1: Example transformations of a news headline using the LLaMA-3-7b-chat model.

The final augmentation action **Ra**, is a function that randomly selects a headline from the training split (ignoring headlines within the base news headline $\mathcal{N}$). This acts similarity to randomly sampling negatives in a traditional contrastive learning mechanism.

Our augmentation stochastic transformation $T : \mathcal{N} \rightarrow (s, \hat{\mathcal{N}})$, generates augmented news headlines defined fully in Appendix A. However, the intuition is quite straightforward. For each news headline, we can generate an augmentation by: 1) Randomly sample the number of headlines within the augmented news headline ($\hat{\mathcal{N}}$) according to the global distribution. 2) According to $P_{actions}$, randomly sample a augmentation action. 3) For each sampled augmentation action, perform that action. Note that the actions **Re**, **S**, and **N** each randomly sample a headline from the base news headline ($\mathcal{N}$), and use that to create an augmented headlines ($\hat{\mathcal{N}}$). 4) Randomly shuffle the order of the augmented headlines in $\hat{\mathcal{N}}$.

In our experiments we set $P_{actions}$ such that: $P(\mathbf{Re}) = 0.05$, $P(\mathbf{S}) = 0.025$, $P(\mathbf{N}) = 0.05$, and $P(\mathbf{Ra}) = 0.775$. These values were used because augmented news headlines produced a similarity score distribution with a high skew to negative scores (as common in contrastive learning), while not overly-depending on negative actions networks. We leave finetuning this probability distribution as a task for future work.

3. Generating Similarity Scores For each augmented news headline $\hat{\mathcal{N}}$, we calculate the similarity score $S(\mathcal{N})$ using a logarithmic weighting function:

$$S(\hat{\mathcal{N}}) = \ln\left(1 + \frac{\sum_{a \in \hat{\mathcal{N}}_A} Sim(a)}{S_{\max}} \cdot (e-1)\right) \quad (2)$$

where a is an augmentation action within the list of augmentation action tuple $\hat{\mathcal{N}}_A$, $S_{\max}$ is the maximum possible total score to normalize the sum to the range $[0, 1]$, and $Sim(.)$ is the function mapping each augmentation action to its corresponding similarity score, such that:

$$Sim(\textbf{Re}) = 1.0, \quad Sim(\textbf{S}) = 0.5, \quad Sim(\textbf{N}) = 0.0, \quad Sim(\textbf{Ra}) = 0.0$$

Intuition: The goal of generating a similarity score is to create a metric between 0.0 and 1.0 that measures how similar a news headline is semantically to its augmentation. When comparing two headlines, we assign high similarity if they are rephrased but semantically identical to each other (**Re**), medium similarity if they are slightly semantically-shifted (**S**), and low similarity if they are semantic opposites (**N**) or completey different (**Ra**).

	Re	**S**	**N**	**Ra**	$S(\hat{\mathcal{N}})$
$\hat{\mathcal{N}}_1$	15	1	0	15	1.00
$\hat{\mathcal{N}}_2$	5	3	1	21	0.53
$\hat{\mathcal{N}}_3$	1	4	4	17	0.29
$\hat{\mathcal{N}}_4$	0	0	1	26	0.00

Table 2: List of augmentation actions from a base news headline, and their accompanying similarity score.

A simple approach to generating a similarity scores between a news headline and its augmentation could be to take the simple mean of all of the augmentation action scores. However, if we observe that two news headlines each have a headline that is semantically identical but just reworded, then we want to take note that those news headlines are so similar. Equation 2, skews the similarity scores such that actions with higher similarity scores have an exponentially larger affect in news headline similarity, than semantically different actions. An example of similarity scores is outlines in table 2. There, we see that if we have an augmented news headline, $\hat{\mathcal{N}}_{\infty}$, that has 15 semantically identical headlines to the base news headline, then the similarity score should be very high. Furthermore, $\hat{\mathcal{N}}_{\triangle}$ is a headline with one semantically negated headline from the original, and the rest are completely disjoint headlines, and so it has a very low semantic similarity.

3.2 Weighted Self-Supervised Contrastive Learning (WSSCL)

Now that we have generated augmented news headlines from training set of anchor headlines, and we have given similarity scores to each of these anchor-augmentation news headlines, we can proceed to generating our news headline similarity embedding space through a weighted self-supervised contrastive learning approach.

Our embedding space optimization task is inspired by Supervised Contrastive Learning Khosla et al. [2021], but is augmented to allow for regressive similarity measurements between anchor and augmented projections instead of binary positive / negative labels. Our representation learning framework consists of 3 sections, the **Encoder Network**, the **Projection Network**, and the **Classification Networks**:

Encoder Network: $e = Enc(x)$ is a LLaMA-3 AI [2024] 7 billion parameter chat model. It was fine-tuned to predict market movement direction (*Fall*, *Neutral*, or *Rise*) from the NIFTY dataset Raeid et al. [2024]. Additional details of SFT implementation are available from Saqur [2024]. news headlines are tokenized and propagated through the encoder network, and the mean values from the last hidden layer are returned, such that $e = \text{Enc}(x) \in \mathbb{R}^{D_E}$. e is then normalized to a hypersphere, which in our implantation had dimensions of 4096.

Projection Network: $p = Proj(e)$ is a feedforward neural network with a single hidden layer, and a shape of (4096, 256, 128), and a single ReLU nonlinearlity unit. The role of this network is to project embeddings e into our embedding space. After projection, the output values are again normalized. We found negligible effects on the quality of the embedding space by increasing the complexity of the projection network.

Classification Networks: $Class_{Proj}(p)$, $Class_{LLM}(e)$ and $Class_{Both}(p, e)$, are tasked with classifying the market movement as rising, falling or neutral. $Class_{Proj}$ takes the projections from the embedding space as an input and $Class_{LLM}$ takes the final hidden states from the encoder LLM. $Class_{Both}(p, e)$ takes both projection and LLM embeddings as inputs. Training of the classification networks is done after the projection network is optimized. Note that for training of the classification networks all augmentations are discarded, and our classifiers are optimized on real news headlines only.

The optimization task we define for our projection network are defined the Weighted Similarity Contrastive Loss (Equation 3).

$$\mathcal{L}_{\text{WSCL}} = \frac{1}{|\mathcal{D}_{newsheadlines}|} \sum_{i=1}^{N} \sum_{j=1}^{M_i} \left[s_{ij} \cdot d_{ij}^2 + (1 - s_{ij}) \cdot \max(0, \delta - d_{ij})^2 \right], \tag{3}$$

Where, N: Total number of anchor news headlines in a batch, M_i: Number of augmented samples for anchor i, $d_{ij} = ||\mathbf{p}_i - \mathbf{q}_{ij}||_2$, $s_{ij} \in [0, 1]$: Similarity score between the anchor and augmented embeddings, and δ is the hyperparameter defining the contrastive margin.

The proposed loss ($\mathcal{L}_{\text{WSCL}}$) extends the classical triplet loss by incorporating a fuzzy similarity score $s_{ij} \in [0, 1]$, enabling a more nuanced handling of relationships between anchor and augmented samples. This formulation draws inspiration from the traditional triplet loss introduced by Schroff et al. [2015]. in FaceNet, which minimizes the distance between anchor-positive pairs while maximizing the distance between anchor-negative pairs using a fixed margin. By replacing binary labels with continuous similarity values, $\mathcal{L}_{\text{WSCL}}$ facilitates a finer gradient flow and captures graded relationships, making it particularly suitable for tasks involving regressive or weighted similarity measures.

The **pull loss** term, $s_{ij} \cdot d_{ij}^2$, minimizes the distance between anchor and augmented embeddings when s_{ij} is high (e.g., $s_{ij} \approx 1.0$). Conversely, the **push loss** term, $(1 - s_{ij}) \cdot \max(0, \delta - d_{ij})^2$, increases the distance between embeddings when s_{ij} is low (e.g., $s_{ij} \approx 0.0$), ensuring proper separation within the embedding space.

In addition to $\mathcal{L}_{\text{WSCL}}$, the Continuously Weighted Contrastive Loss (CWCL) proposed by Srinivasa et al. [2023] is another approach for weighted similarity learning. Unlike $\mathcal{L}_{\text{WSCL}}$, CWCL uses cosine similarity instead of Euclidean distance and incorporates a softmax normalization across all pairs in the batch to enforce global consistency. The CWCL loss is defined as:

$$\mathcal{L}_{\text{CWCL}} = -\frac{1}{|\mathcal{D}_{newsheadlines}|} \sum_{i=1}^{N} \sum_{j=1}^{M_i} s_{ij} \cdot \log \frac{\exp(d_{ij}/\tau)}{\sum_{k=1}^{M_i} \exp(d_{ik}/\tau)}, \tag{4}$$

Where τ is the temperature scaling parameter that controls the sharpness of the distribution. CWCL allows for fine-grained alignment of embeddings by normalizing similarity scores within the batch, providing a complementary perspective to the pull-push mechanics of $\mathcal{L}_{\text{WSCL}}$.

Both approaches aim to improve the representation of graded relationships in embedding spaces but differ in their distance metrics and weighting strategies. In Section 4, we explore each loss function and measure which one performs better on our evaluation tasks.

It is notable that for the WSSCL task, the ground truth market direction corresponding to the news headline's day is not used at all in clustering. The ground truth market direction is saved only for our evaluation tasks (see subsection 3.3). This is so we can measure if the self-supervised task, optimized only for similarity inherently encodes market direction features, without giving them specifically. This lends credence to the idea that through WSSCL information on markets are created.

3.3 EVALUATING SIMILARITY SPACE INFORMATION RICHNESS

Metrics **1) Geometric K-Nearest Neighbors (g-KNN)** evaluates the quality of local label distributions by measuring the entropy of the labels among the k-nearest neighbors of each data point, averaged over the entire dataset. This entropy-based measure provides insights into the local clustering structure of the embedding space. Lord et al. [2018] **2) Nearest Neighbor Accuracy** assesses the proportion of data points whose closest neighbor shares the same category label, providing a direct measure of clustering performance. **3) Kullback-Leibler (KL) Divergence** measures the difference between the local label distribution among the k-nearest neighbors and the global label distribution, indicating the extent to which local clusters differ from random chance Shlens [2014]. **4) Jensen-Shannon Divergence (JSD)** offers a symmetric and bounded evaluation of the similarity between local and global label distributions, enhancing interpretability. These metrics are widely recognized in the literature for their effectiveness in quantifying clustering quality and information richness in embedding spaces Lin [1991].

4 EXPERIMENTAL RESULTS AND INTERPRETATIONS

To analyze the effectiveness of ContraSim we perform two experiemnts. 1) We train a classification network to predict rising, neutral, or falling markets for each provided news headline. We run ablations in which the classification network predicts with only the similarity space embeddings, only the news headline embeddings or both, and compare their performance with accuracy and F1 score. 2) We measure the density of similiarity embeddings by measuring whether ContraSim inherently clusters news headlines of similar market movement together (without knowledge of ground truth market movement) by using g-KNN as a metric.

4.1 DATASETS

For each of these experiments, we compare results on 3 datasets: NIFTY-SFT Saqur et al. [2024], BigData22 Soun et al. [2022], and the IMDB review dataset Maas et al. [2011]. A full analysis of this is outlined in Table 3. **NIFTY-SFT** Saqur et al. [2024] is the collection of WSJ headlines The Wall Street Journal [2024] collected and concatenated together alongside the movement of the US equities market (ticker: $SPY) for the corresponding day. **BigData22** Soun et al. [2022] likewise is a financial news headline dataset, but news headlines are composed of tweets as apposed to WSJ headlines. Finally, we evaluate with the **IMDB review** dataset, which is a collection of human-written reviews for a list of movies alongside the movie's overall review score. An extended analysis of the datasets used is available in Appendix D.

For the IMDB review example, we define a news headline as the concatenated movie reviews, and the prediction task into *Low* (0.0 - 5.5 stars), *Medium* (5.6 - 7.5 stars) and *High* (7.6 - 10.0 stars). We evaluate ContraSim on this dataset to assess its generalizability to orthogonal tasks beyond financial domain prediction.

Name of the Dataset	Problem Domain	Headlines	Days/Movies	Date Range
NIFTY-SFT	Financial Headlines	-	-	-
BigData22	Financial Tweets	-	-	-
IMDB Review	Movie Reviews	-	-	-

Table 3: Summary of the datasets used in the experiments, including their problem domain, the number of headlines (or reviews), the number of days covered, and the date range.

4.2 Results

Table 4 shows that a conjunction of projection, and LLM embeddings are better able to classify news headlines as rising, neutral, or falling when both similarity space projections, and LLM final layer embeddings are used. Using this conjunctive method we achieve a balanced accuracy of .3774%, a 13% increase on the baseline, and a 7% increase on the model using only the LLM embeddings. The model trained only on the projection did worse, just marginally beating the baseline.

Model	**NIFTY-SFT**	**BigData22**	**IMDB**
Baseline	.3333 / .3333	.5000 / .5000	.3333 / .3333
$Class_{Proj}$	.3434 / .3389		
$Class_{LLM}$	.3522 / .3833		
$Class_{Both}$	**.3774 / .4670**		

Table 4: Accuracies and F1 scores (Accuracy / F1 Score) for classification models across datasets. The NIFTY dataset was subsetted to achieve a (33%, 33%, 33%) split.

Table 5 displays embedding space density metrics for a baseline, and our similarity space projection. We observe an increase in clustering accuracies in g-KNN, and KNN, indicating that in the process of ContraSim augmentation and self-supervised contrastive learning, the projection model was able to map points of homogeneous market direction to closer points in space. However, we observe that the projection network actually does worse in KL-Divergence and JSD over the baseline.

Dataset	**Model**	**g-KNN (k=5)**	**KNN (k=5)**	**KL-Divergence**	**JSD**
NIFTY-SFT	Baseline	.5916	.4668	.3539	.1054
	$\mathcal{L}_{\text{CWCL}}$	.7647	.4732	**.3821**	**.1164**
	$\mathcal{L}_{\text{WSCL}}$	**.7219**	**.5205**	.3740	.1144
BigData22	Baseline	.7951	.5506	.1499	.0452
	$\mathcal{L}_{\text{CWCL}}$	**.9084**	**.7101**	.2030	.0607
	$\mathcal{L}_{\text{WSCL}}$	.8590	.5507	**.2246**	**.0640**
IMDB	Baseline	.7456	.5781	.2919	.0818
	$\mathcal{L}_{\text{CWCL}}$	.7626	**.7500**	**.3957**	**.1120**
	$\mathcal{L}_{\text{WSCL}}$	**.8252**	.6875	.3024	.0908

Table 5: Comparison of Baseline and Projection models across datasets and evaluation metrics. Note that finding true baseline values for these metrics on unbalanced sets of labels is nontrivial and out of scope for this paper. As a result, estimated baseline values are the mean of 1000 cases of randomly distributed points following the respective label splits for each dataset. The best results are in **bold**.

4.3 Discussions

[**RS**: TODO: @NV UPDATE THE CONCLUSON]

We conclude that by using ContraSim to generate a similarity space, and using that similarity space as a feature for supervised learning, we generate domain information that was not there originally. This is also reinforced in the structure of the similarity space itself, as we some have evidence that the method is able to clump homogeneous market movement days closer together than by chance.

4.4 Training Details

The Projection Network was trained for 50 epochs using $\mathcal{L}_{\text{combined}}$ loss. Hyperparameter search was done in three phases. First, a small set of learning rates (0.1, 0.001, 0.0001), and gamma decay values (0.95, 0.90, 0.85) where optimized for.

The Classification Networks were all optimized in very similar ways. Like the projection network we performed a sweep on learning rate and gamma decay. Cross entropy loss was used, and projection values that were used as inputs to $Class_{Proj}$, came from the best performing projection model based on the test set g-KNN (k=5) scores.

5 Future Work

For future work, we aim to expand ContraSim beyond financial data by testing it on other domains such as healthcare, legal, and social media datasets. This will help assess the model's generalizability across diverse text types and semantic contexts. Additionally, we plan to incorporate more recent language models, like GPT-4 or Meta LLaMA 3, to enhance the embedding quality and clustering performance. Exploring these models' fine-tuning capabilities in unsupervised financial forecasting could further strengthen ContraSim's ability to handle complex text data. We could also incorperate other Contrastive Learning features such as hard negative mining, and dynamic temperature scaling.

Ethics Statement [**RS**: here explicate why the 'S' step (i.e. shifted semantic headline) is not harmful and later we refer to this paragraph for rebuttling the reviewer who raised ethics violation because of fake news assupmption]

Reproducibility Statement The authors of this paper ensure reproducibility through 1) The accurate and clear descriptions of methods used, specifically in the training details and methods sections of the text, 2) The Use of only public models and datasets (NIFTY), and 3) Providing source code in the supplemental materials (see attached).

REFERENCES

Swati Aggarwal, Deepak Kumar, Sandeep Dahiya, and Nisha Kaur. Spam detection using machine learning and deep learning techniques. *International Journal of Advanced Computer Science and Applications*, 13 (2), 2022.

Meta AI. Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/, 2024. Accessed: 2024-05-21.

Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3): 307–327, 1986.

George EP Box and Gwilym M Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. *Proceedings of the 17th Pacific-Asia conference on knowledge discovery and data mining (PAKDD)*, pp. 160–172, 2013.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

Xiaoxue Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*, pp. 2327–2333, 2015.

Robert F Engle and Clive WJ Granger. Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pp. 251–276, 1987.

Thomas Fischer and Christopher Krauss. Stock market prediction using deep learning models. *Journal of Business Research*, 96:456–467, 2018.

James D Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, pp. 357–384, 1989.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Charles C Holt. Forecasting seasonals and trends by exponentially weighted averages. Technical report, Office of Naval Research, 1957.

Zhanxing Hu, Wenyuan Liu, Jiang Bian, Hao Liu, and Yajuan Zheng. A deep learning approach for stock market prediction based on financial news. In *Proceedings of the 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pp. 119–124, 2018.

Ting Jiang et al. Scaling sentence embeddings with large language models. *ArXiv*, 2023. URL https://consensus.app/papers/scaling-sentence-embeddings-large-language-models-jiang/f5c6fb92876653a1a2cb0b6f3eac1fe2/?utm_source=chatgpt.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. In *Journal of Documentation*, volume 28, pp. 11–21. Emerald Group Publishing Limited, 1972.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

Corentin Kervadec et al. Unnatural language processing. *ArXiv*, 2023. URL https://consensus.app/papers/language-processing-language-models-handle-kervadec/66c2591139b85ce29f4303270f7ae5b4/?utm_source=chatgpt.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2021. URL https://arxiv.org/abs/2004.11362.

J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1): 145–151, 1991. doi: 10.1109/18.61115.

Qing Liu, Junjie Liu, and Xiaolin Ren. Financial news prediction using pre-trained language models. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pp. 196–210, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Warren M. Lord, Jie Sun, and Erik M. Bollt. Geometric k-nearest neighbor estimation of entropy and mutual information. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(3), March 2018. ISSN 1089-7682. doi: 10.1063/1.5011683. URL http://dx.doi.org/10.1063/1.5011683.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Alec Radford, Jeffrey Wu, Dario Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

S. Raeid, R. Frank, K. Kato, and N. Vinden. Nifty financial news headlines dataset, 2024. Manuscript under review.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

Claude Sammut and Geoffrey I. Webb (eds.). *TF–IDF*, pp. 986–987. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_832. URL https://doi.org/10.1007/978-0-387-30164-8_832.

Raeid Saqur. What teaches robots to walk, teaches them to trade too – regime adaptive execution using informed data and llms, 2024. URL https://arxiv.org/abs/2406.15508.

Raeid Saqur, Ken Kato, Nicholas Vinden, and Frank Rudzicz. Nifty financial news headlines dataset, 2024. URL https://arxiv.org/abs/2405.09747.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823. IEEE, 2015.

Ömer Faruk Sezer and Murat Ozbayoglu. Tensor-based learning for predicting stock movements. *IEEE Access*, 6:59125–59141, 2018.

Jonathon Shlens. Notes on kullback-leibler divergence and likelihood, 2014. URL https://arxiv.org/abs/1404.2000.

Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pp. 1–48, 1980.

Sumeet S. Singh. Analyzing transformer dynamics as movement through embedding space, 2023.

Yejun Soun, Jaemin Yoo, Minyong Cho, Jihyeong Jeon, and U Kang. Accurate stock movement prediction with self-supervised learning from sparse noisy tweets. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 1691–1700. IEEE, 2022.

Rakshith Sharma Srinivasa, Jaejin Cho, Chouchang Yang, Yashas Malur Saidutta, Ching-Hua Lee, Yilin Shen, and Hongxia Jin. Cwcl: Cross-modal transfer with continuously weighted contrastive loss, 2023. URL https://arxiv.org/abs/2309.14580.

The Wall Street Journal. The Wall Street Journal, 2024. https://www.wsj.com.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Chun-I Tsai and Yin-Jing Wang. Forecasting stock returns with ensemble learning and sentiment analysis. In *2016 International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, pp. 1–6. IEEE, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. Learning paraphrastic sentence embeddings from back-translated bitext. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 274–285, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1026. URL https://aclanthology.org/D17-1026.

Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. Hybrid deep sequential modeling for social text-driven stock prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 1627–1630, 2018.

Yan Xi, Qirong Mao, and Ling Zhou. Weighted contrastive learning using pseudo labels for facial expression recognition. *The Visual Computer*, 39:1–12, 08 2022. doi: 10.1007/s00371-022-02642-8.

Yongchao Xu, Seth B Cohen, Tianqi Zhao, and Amrita Amar. Sentiment analysis for stock price prediction using deep learning models. In *Proceedings of the International Conference on Web Information Systems Engineering (WISE)*, pp. 315–322, 2018.

Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1970–1979, 2018.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*, 2020.

Appendices

Appendix Contents

A HEADLINE TRANSFORMATIONS

Algorithm 1: Stochastic news headline Augmentation Transformation T

Input: Original news headline $\mathcal{N} = (h_1, h_2, \ldots, h_m)$
Input: Action distribution P_{actions} over actions $\{\mathbf{Re}, \mathbf{S}, \mathbf{N}, \mathbf{Ra}\}$
Output: Augmented news headline $(\hat{\mathcal{N}}, s)$ with similarity score s

Sample $n \sim$ Distribution of news headline lengths in corpus
Initialize $\hat{\mathcal{N}} \leftarrow \emptyset$, $S \leftarrow 0$
for $i \leftarrow 1$ **to** n **do**
  Sample $a_i \sim P_{\text{actions}}$
  if $a_i \in \{\boldsymbol{Re}, \boldsymbol{S}, \boldsymbol{N}\}$ **then**
    Sample headline $h \sim \mathcal{N}$
  else if $a_i = \boldsymbol{Ra}$ **then**
    Sample random headline $h \sim$ corpus
  end if
  if $a_i = \boldsymbol{Re}$ **then**
    $h' \leftarrow \text{Reword}(h)$
    $S \leftarrow S + 1.0$
  else if $a_i = \boldsymbol{S}$ **then**
    $h' \leftarrow \text{SemanticShift}(h)$
    $S \leftarrow S + 0.5$
  end if
  else if $a_i = \boldsymbol{N}$ **then**
    $h' \leftarrow \text{Negate}(h)$
    $S \leftarrow S + 0.0$
  end if
  else if $a_i = \boldsymbol{Ra}$ **then**
    $h' \leftarrow h$
    $S \leftarrow S + 0.0$
  end if
  Append h' to $\hat{\mathcal{N}}$
end for
Shuffle $\hat{\mathcal{N}}$
Compute similarity score $s \leftarrow S/n$
return $(\hat{\mathcal{N}}, s)$

B NEWS HEADLINE SIMILARITY EXAMPLES

C MEASURING INFORMATION DENSITY OF NIFTY AND RAEID MODELS WITH IG-CLUPE

In this section, we do an analysis of prompt embeddings to provide evidence for the efficacy of RAEiD finetuning, and the information density of NIFTY over other Flare stock movement datasets. When processing prompts, transformer models like LLaMA-2 Touvron et al. [2023] and LLaMA-3 AI [2024] produce large-dimensional vectors that capture the structure and semantic features. Consequentially, we find that prompt-

	Original Headline “Johnson & Johnson to Buy Surgical Robotics Maker Auris”
	Reworded Augmentation
System	*Please reword this headline for me, preserving the exact semantic meaning perfectly. Your returned headline should contain the exact information with no meaning added or subtracted, but just rephrased. Please generate the headline, and return only that with no other text. Thanks.*
User	"Johnson & Johnson to Buy Surgical Robotics Maker Auris"
Responses	1. Auris Acquired by Pharmaceutical Giant Johnson & Johnson 2. Auris Acquired by Pharmaceutical Giant Johnson & Johnson in Deal Involving Surgical Robot Technology 3. Auris Acquired by Pharmaceutical Giant Johnson & Johnson in Deal Involving Surgical Robot Manufacturer 4. Auris Manufacturer of Surgical Robots Acquired by Johnson & Johnson 5. Auris Medical Acquired by Pharmaceutical Giant Johnson & Johnson in Deal Involving Surgical Robotics Manufacturer
	Semantic-Shift Augmentation
System	*Please modify this headline slightly, so it is about something related but different. If the headline is good news, ensure it remains good news, and if it is bad news, ensure it remains bad news. Please generate the headline, and return only that with no other text. Thanks.*
User	"Johnson & Johnson to Buy Surgical Robotics Maker Auris"
Responses	1. Abbott Laboratories Acquires Medical Imaging Specialist Siemens Healthineers 2. J&J Acquires Medtech Firm Verb for Improved Cancer Treatment Options 3. J&J Acquires Orthopedic Implant Company Zimmer Biomet in $14 Billion Deal 4. J&J Acquires Orthopedic Implant Company DePuy Synthes for $21 Billion 5. J&J Acquires Medtech Company Verb Surgical in $3 Billion Deal
	Negative Augmentation
System	*Please reword this headline for me such that the information is the same except that it now is about the opposite meaning. Please generate the headline, and return only that with no other text. Thanks.*
User	"Johnson & Johnson to Buy Surgical Robotics Maker Auris"
Responses	1. Auris to Sell Off Stake in Surgical Robotics Business to Johnson & Johnson 2. Auris Abandons Plans to Acquire Surgical Robot Business from Johnson & Johnson 3. Auris to Sell Majority Stake to Rival of Johnson & Johnson’s Surgical Robot Division 4. Auris Acquires Surgical Robotics Leader Johnson & Johnson 5. Auris Abandons Plans to Acquire Surgical Robotics Giant Johnson & Johnson

Table 6: Rephrasing, slight ablation, and negative modification of the headline "Johnson & Johnson to Buy Surgical Robotics Maker Auris." Each augmentation displays the system prompt, user-provided headline, and model-generated responses listed with numbers.

embeddings localized in a group contain more similar semantic features than those of sentences of further distance in the embedding space Wieting et al. [2017]. We propose *Information Gain in Clustered Prompt Embeddings* (IG-CluPE), that measures regional prompt similarity by measuring homogeneity of clustered embedding. Using IG-CluPE, we investigate to what degree expert models is able to group prompts with the same direction of market movement together. We show that this information density approach provides insights to both the efficacy of each finetuning stage in RaEID, and to provide evidence that NIFTY encapsulates a

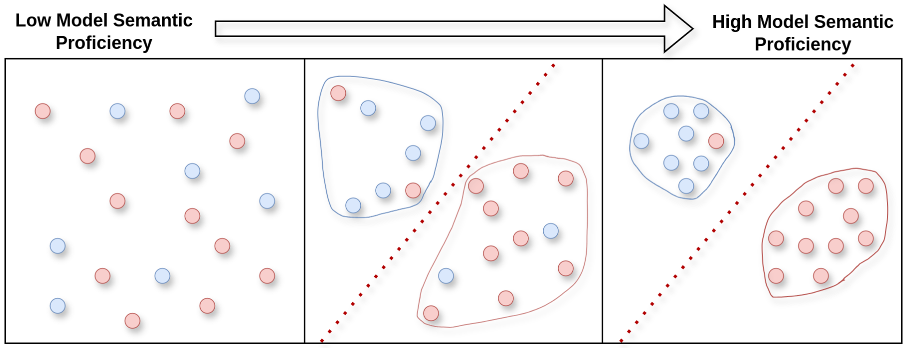


Figure 2: **Information Gain in Clustered Prompt Embeddings (IG-CluPE):** A novel method of measuring a LLM's ability to capture rich semantic contextualization of a corpus of text prompts with corresponding classifications. Prompt embeddings are extracted from outputs of the last-hidden-layer of transformer models to create an embedding space optimized for linear separability of points from each class. The effectiveness of a model's ability to group points with similar features together is measured through t-SNE clustering and information gain.

richer set of features than ACL18 Xu & Cohen [2018], BigData22 Soun et al. [2022], or CIKM18 Wu et al. [2018].

Contributions Below we highlight the contributions of this embedded prompt analysis:

1. **Information Gain in Clustered Prompt Embeddings (IG-CluPE)** : Inspired by from insight in the optimization of decision trees, we outline IG-CluPE: *an algorithm that measures cluster similarity of categorical prompts*. We verify that large language models trained on classification tasks, like predicting the direction of market movement, cluster points of similar financial features together through clustered information gain and an analysis of cluster similarity.
2. **RaEID Information Gain**: Using IG-CluPE, we measure information gain between a pretrained LLaMA-2 model, an UnREAL model with only SFT, and an UnREAL model with both SFT and RLMF.
3. **Dataset Information Richness**: We measure IG-CluPE on ACL18, BigData22, and CIKM18, using information gain as evidence that NIFTY is more information dense, extending the capabilities of SOTA stock price movement datasets.

C.1 IG-CluPE: Information Gain in Clustered Prompt Embeddings

We introduce IG-CluPE, a method of measuring the density of information in classification models. In this section, we show that outputs from LLaMA-2 and LLaMA-3 last-hidden layer can be used as a method of investigating localized information richness within a embedding space in a prompt classification task. IG-CluPE is acquired as a result of the finetuning task in a classification task, and does not need to be created through additional training steps. We find that without explicit guidance, like those used in contrastive learning, localized prompt structures reveal themselves within an embedding space through optimizing a model for a classification task alone. This phenomenon is shown in detail in subsections C.2 and C.3.

Generating prompt embeddings with LLMs is a rich and vibrant field of study, owing to their usefulness in knowledge representation in a field dominated by "black box" algorithms. Examples like Jiang et al. (2023) Jiang et al. [2023] enhancing sentence embeddings through in-context learning, Kervadec et al. (2023) Kervadec et al. [2023] analyzing responses to machine-generated prompts, reveal significant differences in both model responses and network processing pathways. However, each of these methods require training regiments to generate their rich embeddings. Our method of embedding analysis is unique that it is a product of learning intrinsically.

S. Singh (2023) Singh [2023] explores the intelligent behaviors exhibited by Transformer-based language models through the lens of their movement in embedding space. It establishes a theory that maps these intelligent behaviors to paths in embedding space, which are traversed by the transformer during inference, with learning occurring by assigning higher probabilities to paths representing intelligent actions. This study highlights the benefits of viewing learning through the lens of traversing an embedding space. However, since this approach focuses on creating embeddings through a transformer's parameter space, approaches like these lose the ability to parse inter-prompt semantic similarities.

Generating information gain of an embedded space using IG-CluPE is outlined in these steps:

1. Embedding Generation: We feed through each tokenized prompt (x_p) through our LLaMA model, extracting and saving the outputs from the final hidden layer of the transformer as prompt embeddings.

2. Prompt Clustering: Once embeddings are generated for all prompts, we use *t-distributed Stochastic Neighbor Embedding* (t-SNE) van der Maaten & Hinton [2008] to cluster all prompts. For purposes of visualization we also use HDBSCAN Campello et al. [2013] for creating cluster figures in Cartesian space.

3. Information Gain Measurement: We measure the information gain of clustering each prompt with equations 5 - 8, where L is a set of M tags $(l_1, l_2, \dots, l_M)$, T is a multiset of N classification tags such that each element $t \in T$ is also in L, and $\{P_1, P_2, \dots, P_K\}$ is a partition of T into K clusters.

$$p(l,T) = \frac{|\{i \in T : \text{label of } i = l\}|}{|T|} \tag{5}$$

$$H(T) = -\sum_{l \in L} p(l,T) \log_2 p(l,T) \tag{6}$$

$$H_C(P) = \sum_{k=1}^{K} \frac{|P_k|}{|T|} H(P_k) \tag{7}$$

$$IG = H_C(P) - H(T) \tag{8}$$

The intuition behind using a last-hidden-layer embedding clustering-based approach to measure information richness is rooted in the optimization processes of classification models. By analyzing the embeddings from the final hidden layer of a neural network, we can assess how well the model captures and discriminates between different classes of data. Clustering these embeddings allow us to both qualitatively and quantitatively evaluate the separability and density of the data representations, reflecting the model's ability to generalize and its sensitivity to various financial features. This approach not only offers insights into the model's internal representations but also allows us to generalize which datapoints share pertinent features to stock movement prediction.

By using the last-hidden-layer of a transformer architecture, preceding a single-layer fully connected neural module, we ensure that during training the model is optimized for linear separability of the last-hidden-layer embedding space. Therefore, we pose that in an optimized model a single data point has an increased probability of being surrounded by data points of the same class, as compared to a worse performing model.

We borrow a technique of measuring the degree of cluster homogeneity through information gain in decision tree optimization, described in equations 5 - 8. When optimizing a decision tree for a classification task, we

̈splitän initial set of datapoints based on differing features of each point. For example, in classifying a set of apples as *McIntosh* or *Granny Smith*, we may choose to based on whether each apple is red or green, or if each apple is above 200g or below 200g. Decision trees are optimally split by choosing the feature that creates the highest amount of information gain post clustering. Intuitively we can see that colour is likely to be a more informationally dense feature when classifying *McIntosh* or *Granny Smith* apples as compared to weight.

IG-CluPE leverages insights from decision tree optimization and flips the process in the opposite direction. Instead of using information gain to guide method of clustering, with IG-CluPE clustering is provided through t-SNE localized clustering, and is used to measure information gain of the model.

Although we state that a model's IG-CluPE score is proportional of the model's ability to perform a downstream classification task it was optimized for, we find that using **IG-CluPE has a set of marked advantages** over only using classification accuracy for model evaluation. Clustering prompt allows us to better interpret model decision, and lets us view which prompt features the model finds useful in prompt classification. Whereas viewing model performance solely through the lens of classification accuracy groups each prompt into one of N categories, IG-CluPE allows us to peer into the prompt space and visualize how the model groups similar points. We can look at false negatives and see which other prompts are closest to that prompt in the embedding space.

Model information density analysis through IG-CluPE is additionally insightful in the domain of comparing the efficacy of similar LLM models, or when comparing the information density of similar datasets. A IG-CluPE can guide model design by peering into the inner workings of the model and identifying weak points. For example, in the context of semantic classification, if a model predominantly groups prompts of classes *HAPPY* and *EUPHORIC* together, we could tweak training methodology to include more cases of these classes in the dataset. Then another embedding space can be created, and results compared. Additionally, we can also look at how information density in the context of identically trained models embedding prompts of similar datasets. A dataset with a modified/additional set of features can guide the models ability to correctly classify text phrases. A clustered embedding space for each dataset can highlight how our model utilize changes in the feature set.

In sections C.2 and C.3, we test both of these cases by using IG-CluPE to measure information richness of a base pretrained LLaMA-2 model, an UnREAL model trained with only SFT, and a full UnREAL model with SFT and RLMF. We also use IG-CluPE to measure the information richness of ACL18, BigData22, CIKM18, and NIFTY. In both of these experiments we analyze the clustered embedding space, and describe both qualitative and quantitative results, revealing a model's semantic structures and identifying strengths and points of weakness. All prompt clusters for both experiments were done with 3000 iterations of t-SNE with a minimum cluster size of 3 on the test section of each of the datasets of interest.

C.2 RAEID INFORMATION GAIN

In this section we outline an IG-CluPE approach to measuring information richness of various training stages of the UnREAL model. For each stage in the UnREAL training algorithm **??**, we generate prompt embeddings over the entire NIFTY test set. Embeddings are taken as the output of the final transformer layer in the LLaMA-2 encoder architecture. Embeddings are clustered using t-SNE with a minimum cluster size of 5, where the category of each point corresponds to 'rise', 'fall' or 'neutral' stock movement. Below we interpret the IG-CluPE score and take a deeper look into the embedding space structure with qualitative and quantitaive metrics.

Additionally, we measure how the correlation between increased IG-CluPE information gain and classification accuracy of the model. If we find there is a high correlation, then that provides evidence that through finetuning a model for a classification task intrinsically creates clustering of same category points.

Finally, IG-CluPE is predicated on the idea that an increase in information gain is indicative of a better performing model. Only from that axiom, can we explore inter-prompt relationships on the basis that finetuning a model on a classification task creates dense homogenous regions in the embedding space. To explore this relationship we calculate a Pearson's correlation coefficient between both Accuracy and F1 score with IG-CluPE.

RaEID Models Tested: Here we list the models we are interested in testings.

1. LLaMA-2-7b
2. LLaMA-2-7b-chat
3. LLaMA-2-7b-chat + SFT
4. LLaMA-2-7b-chat + SFT + RLMF (UnREAL)

By testing LLaMA-2-7b models from no finetuning to a full UnREAL model, we intend to observe how clustered points change. We measure the shift of prompts of the same market direction clump together through the context of information gain and extend that through a deeper cluster analysis.

Table 7: F1 Score, Accuracy, and Information Gain from Clustered Prompt Embeddings (IG-CluPE) for different stages of training in the RAEiD training algorithm **??** - **??**.

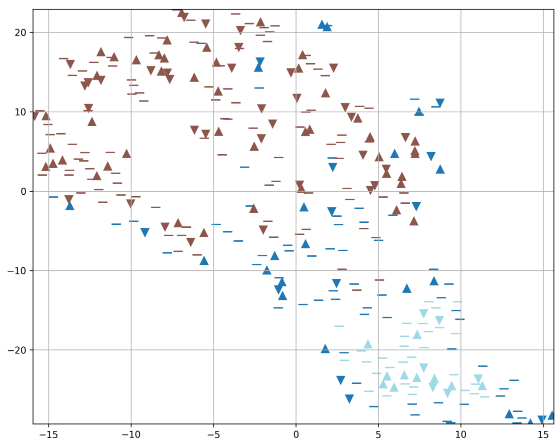
Model	Acc.	F1	IG-CluPE
LLaMA-2-7b	0.23	0.21	0.0117
LLaMA-2-7b-chat	0.27	0.22	0.0835
LLaMA-2-7b-chat + SFT	0.45	0.28	0.0835
LLaMA-2-7b-chat + SFT + RLMF (UnREAL)	**0.71**	**0.72**	**0.1039**

Discussion In Table 7 we observe an increase/maintaining of clustered information gain throughout training. This gives evidence that IG-CluPE is an appropriate metric for measuring information richness of prompt embeddings. We see as the model's classification ability increases, it creates more dense regions of high information within the embedding space. We also observe very similar scores in Table 7 and a very similar embedding structure in Figure 3. Information gain increases considerably post reinforcement learning with market feedback, showing the importance and quality that stage brings to model performance.

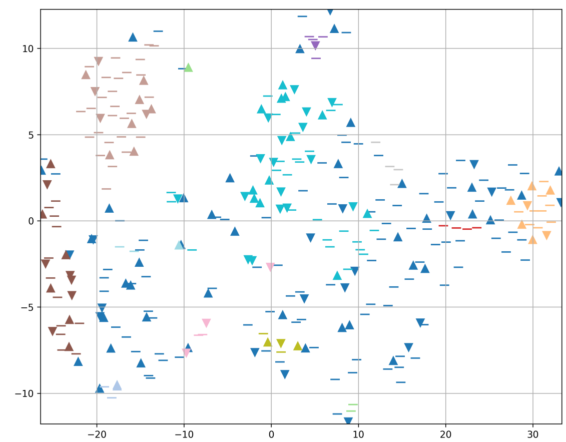
Additionally, we investigate inter-cluster trends highlighted in Tables 10 and 11. In Table 10, we observe as the complexity of training from the base model to UnREAL increases, the prompt embedding space is better clustered in terms of information gained. While the set of unclustered prompts has around a 23%, 60%, 17% split of rising, neutral and falling stock days, post clustering we find specific clusters that contain much less ambiguity. For example, clusters 15, 16, and 17 are in a region of the embedding space that is extremely conducive to neutral stock movement with nearly all stocks in this group being neutral stock movement. However, cluster 9 shows the opposite with no points being labelled as neutral.

In Table 11, we observe the results of tokenizing each news headline and analyzing the most significant words within each cluster. By employing a TF-IDF Jones [1972] algorithm, we identified the top words that characterize each cluster. This analysis reveals interesting trends within the stocks and provides insights into market sentiment. For example, Cluster 3, characterized by words like "Hole," "Crash," and "Johnson," indicates a strong positive trend, reflected in its highest normalized market movement of 1.0000. Conversely, Cluster 11, with prominent words such as "Last," "Place," and "EU," shows a strong negative trend, evidenced by its normalized market movement of -1.0000. Clusters 15 and 13, featuring terms like "Shift," "Glitch," and "Complaint," are situated in the negative movement region, suggesting a downturn in market sentiment.

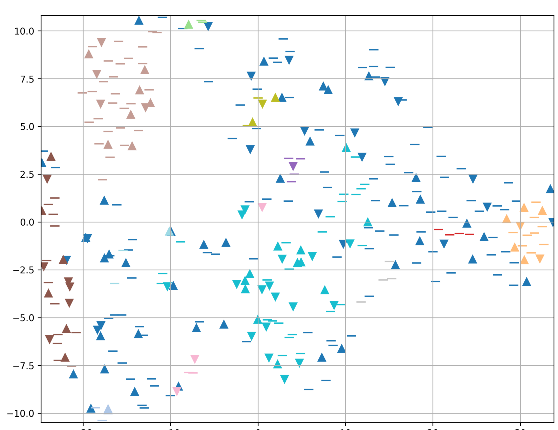
These findings highlight the effectiveness of clustering and TF-IDF analysis in uncovering nuanced patterns and trends in stock movement data, as evidenced by the frequently occurring words in each cluster.



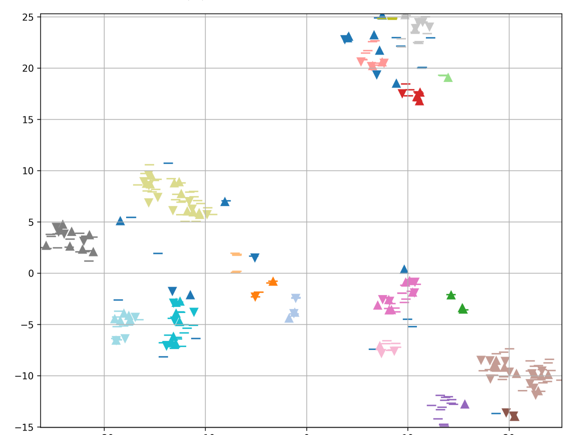
(a) LLama-2-7b



(b) LLaMA-2-7b-chat



(c) LLaMA-2-7b-chat + SFT



(d) LLaMA-2-7b-chat + SFT + RLMF (UnREAL)

Figure 3: Clustered prompt embeddings for each point in NIFTY-test, generated from LLaMA-2-7b models from no finetuning to full RLMF. Prompts embeddings were clustered and generated with the IG-CLuPE algorithm. Symbols ▲, ▼, and **-** corresponds to rising, falling and neutral market movement. Each prompt is clustered in a group that can be differentiated by color with notably dark blue corresponding to outlier prompts belonging to no cluster. With points. An embedding space with the highest information gain is one that groups the most points of the same cardinality together. We observe each step of training increasing the amount of information gain.

C.3 Dataset Information Richness

Here we outline an experiment on information richness in our dataset NIFTY, and we compare it to other SOTA financial movement tasks, ACL18, BigData22, and CIKM18. Where in the previous section we created embeddings on NIFTY from various models in training, here we keep training constant and we change which dataset embeddings are being generated for. The models generating the embeddings will be UnREAL models fintunied on the training set of their respective financial dataset. Then we will generate prompt embeddings for each of our datasets and compare information density with IG-CluPE.

It is notable that ACL18, BigData22, and CIKM18 only contain 2 categories: *Rise* and *Fall*. In order to maintain parity, we omit all *Neutral* prompts from the test set and use the subsetted NIFTY dataset for evaluation.

Table 8: Information Gain (IG), Accuracy, and F1 Score for LLaMA-2-7b-chat across different datasets

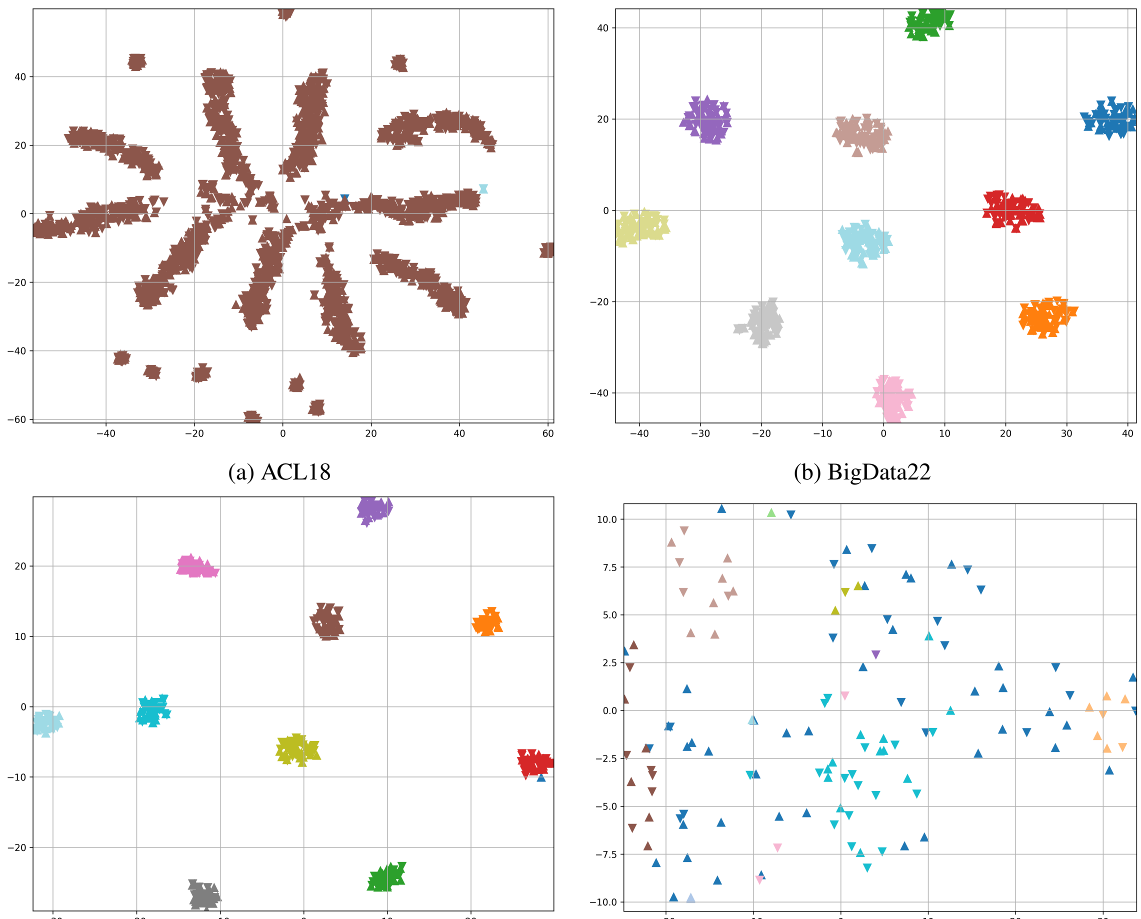
Model	Acc.	F1	IG-CluPE
LLaMA-2-7b-chat + ACL18	0.25	0.20	0.0000
LLaMA-2-7b-chat + BigData22	0.29	**0.29**	0.0061
LLaMA-2-7b-chat + CIKM18	0.27	0.27	0.0107
LLaMA-2-7b-chat + NIFTY	**0.45**	0.28	**0.0997**

Discussion We observe much higher information gain from clustered prompt embeddings in the model that is finetuned and evaluated on the NIFTY dataset. This coincides with a larger accuracy score on the financial movement classification task. We conclude that the NIFTY dataset contains more pertinent information that allows the model to better predict movement in $\$SPY$ prices. Interestingly, we find that ACL18, BigData22, and CIKM18 create extremely dense point clusters, compared to NIFTY, once condensing down to a two dimensional representation. However, these clusters do not have a strong direction homogenity, and thus do not provide much information on the basis of trends in market movement. Thus we conclude that the model finetuned on NIFTY is better able to represent and understand concepts of the factors in daily news and financial data to predict market movement.

Table 9: Pearson correlation coefficients and p-values for the correlation between Accuracy, F1, and IG-CluPE. The evaluation was performed using Pearson correlation tests on the combined data from different stages of training in the RAEiD algorithm and different datasets.

Metric	Pearson Correlation Coefficient	p-value
Accuracy	0.759	**0.029 ****
F1	0.514	0.192

Discussion: In this experiment highlighted in Table 9, we look at Accuracy, F1 and IG-CluPE scores from Tables 7 and 8. We find that within a 95% statistical level of significance that there is a correlation between accuracy of downstream prediction and IG-CluPE score. However, this does not apply to F1 score. The significance between IG-CluPE and accuracy provides evidence that IG-CluPE is a useful tool for measuring a model's effectiveness, and that there is evidence that richer models create richer regions of information within a prompt-wise embedding space. However, we would like to further investigate correlation between IG-CluPE and F1.



(a) ACL18 (b) BigData22

(c) CIKM18 (d) NIFTY

Figure 4: Clustered prompt embeddings for LLaMA-2-7b-chat models finetuned on ACL18, BigData22, CIKM18, and NIFTY datasets. Prompts embeddings were clustered and generated with the IG-CLuPE algorithm. Symbols ▲, ▼, and **-** corresponds to rising, falling and neutral market movement. Each prompt is clustered in a group that can be differentiated by color with notably dark blue corresponding to outlier prompts belonging to no cluster. With points. An embedding space with the highest information gain is one that groups the most points of the same cardinality together. We observe each step of training increasing the amount of information gain.

Table 11: Clusters Ordered by Normalized Market Movement

Cluster	**Market Movement**	**Top 10 Words (TF-IDF)**
3	0.005480 (**1.0000**)	Hole, Crash, Johnson, Suing, Audit, Rubin, Combinator, Jackson, CEOs, Gap
17	0.001771 (**0.1838**)	Shuts, Bank, Refinements, Brokerages, Normalization, Stock, LIBOR, Momentum, Revenue, Ex

Continued on the next page

Cluster	Market Movement	Top 10 Words (TF-IDF)
Outliers	0.001724 (**0.1735**)	Removal, Slot, Critic, GOP, Consulting, Tracy, Cordray, Housing, Plants, Price
4	0.002000 (**0.2343**)	Scotsman, Sushi, Sponsor, Zions, Peace, Economies, Brexit, Tracy, Wage, Euro
12	0.001578 (**0.1414**)	Mr, Funding, Clayton, Results, Amendment, Nominee, Things, Withdraws, Way, Series
5	0.001560 (**0.1375**)	Sailing, Justices, Protections, Powell, CFPB, Produce, Misfires, Resilinc, Baer, Prognos
1	0.001443 (**0.1117**)	Test, Waves, Let, Affects, BofA, Holders, Analyst, Mills, Conflict, Carney
8	0.001375 (**0.0968**)	Standard, Words, Trustee, Transcript, Broker, York, Take, Wynn, Dudley, Interview
7	0.001060 (**0.0275**)	Aid, Exempts, Tracy, Tests, Regulation, Payout, Cyberattack, Tries, Measures, Pressure
2	0.000671 (**-0.0580**)	Emails, Recess, Outcry, Denials, Bipartisanship, Pressure, Pound, LA, Paloma, Las
10	0.000543 (**-0.0861**)	Ambitions, June, Reluctant, FTI, Consulting, Jolt, Rubin, Budget, CFTC, London
16	0.000462 (**-0.1042**)	Burn, Conn, Seritage, Pit, Stability, Hartford, Triggers, Future, Lawmakers, Phone
14	0.000425 (**-0.1122**)	Arbitration, Database, Defections, Peltz, Senators, CFPB, Urges, Control, Sets, Support
6	-0.000108 (**-0.2294**)	State, Collection, Hayashi, Contenders, Coin, Votes, Agenda, Gasoline, Boards, Statement
9	-0.000080 (**-0.2232**)	Solution, Crisis, Conviction, Bondholder, Breitburn, Policy, Team, Deng, Scurria, Bet
15	-0.000500 (**-0.3157**)	Monte, Italy, Shift, Benefits, Worldpay, Control, Glitch, Minutes, Meeting, Roundup
13	-0.001876 (**-0.6186**)	Program, Water, Scaramucci, Complaint, Communications, Database, Director, HNA, Banker, Path
0	-0.003443 (**-0.9632**)	Expectancy, Quarles, Ackerman, Broadbent, Sheet, Senate, CFPB, Arbitration, Czar, Life
11	-0.003610 (**-1.0000**)	Last, Place, EU, Punts, Mooch, Banking, Verlaine, Fees, Consumers, Entertainment

D Datasets and Benchmarks

D.1 NIFTY Dataset

The **N**ews-**I**nformed **F**inancial **T**rend **Y**ield (NIFTY) dataset Raeid et al. [2024] is a processed and curated daily news headlines dataset for the stock (US Equities) market price movement prediction task. NIFTY is comprised of two related datasets, NIFTY-LM and NIFTY-RL. In this section we outline the composition of the two datasets, and comment on additional details.

Dataset statistics Table 12 and Table 13 present pertinent statistics related to the dataset.

Anticipate the direction of the $SPY by analyzing market data and news from 2020-02-06.

(a) Instruction component of a π_{LM} policy query x_q.

date, open, high, • • •, pct_change, macd, boll_ub, boll_lb, rsi_30, • • •, close_60_sma

2020-01-27, 323.03, 325.12, • • •, -0.016, 2.89, 333.77, 319.15, 56.26, • • • , 317.40
2020-01-28, 325.06, 327.85, • • •, 0.0105, 2.59, 333.77, 319.55, 59.57, • • • , 317.78
• • •. • • • •
2020-02-04, 328.07, 330.01, • • •, 0.0152, 1.3341, 333.60, 321.26, • • •, 319.41
2020-02-05, 332.27, 333.09, • • •, 0.0115, 1.7247, 334.15, 321.73, • • •, 319.82

(b) The market's **history** is provided as the past t days of numerical statistics like the (OHLCV) price (in blue) and common technical indicators (in orange) (e.g. moving averages) data.

Figure 5: Breaking down the instruction or prompt prefix, and market context components of a prompt, x_p.

D.1.1 NIFTY-LM: SFT Fine-tuning Dataset

The NIFTY-LM prompt dataset was created to finetune and evaluate LLMs on predicting future stock movement given previous market data and news headlines. The dataset was assembled by aggregating information from three distinct sources from January 6, 2010, to September 21, 2020. The compilation includes headlines from The **Wall Street Journal** and **Reuters News**, as well as market data of the $SPY index from **Yahoo Finance**. The NIFTY-LM dataset consists of:

- **Meta data**: Dates and data ID.
- **Prompt** (x_p): LLM question ($x_{question}$), market data from previous days ($x_{context}$), and news headlines (x_{news}).
- **Response**: Qualitative movement label (x_r) $\in \{Rise, Fall, Neutral\}$, and percentage change of the closing price of the $SPY index.

To generate LLM questions, ($\boldsymbol{x_{question}}$), the authors used the self-instruct Wang et al. [2023] framework and OpenAI GPT4 to create 20 synthetic variations of the instruction below:

> Create 20 variations of the instruction below.
> Examine the given market information and news headlines data on DATE to forecast whether the $SPY index will rise, fall, or remain unchanged. If you think the movement will be less than 0.5%, then return 'Neutral'. Respond with Rise, Fall, or Neutral and your reasoning in a new paragraph.

Where `DATE` would be substituted later, during the training phase with a corresponding date.

Context The key 'context' ($\boldsymbol{x_{context}}$) was constructed to have newline delimited market metrics over the past T ($\approx$ 10) days (N.B. Not all market data for the past days for were available and therefore prompts might have less than 10 days of market metrics.).

Table 14 show the details of financial context provided in each day's sample.

News Headlines ($\boldsymbol{x_{news}}$): Final list of filtered headlines from the aggregation pipeline. The non-finance related headlines were filtered out by performing a similarity search with SBERT model, "all-MiniLM-L6-v2" Reimers & Gurevych [2019]. Each headline was compared to a set of artificially generated financial headlines generated by GPT-4, with the prompt *"Generate 20 financial news headlines"*. Headlines with a similarity score below 0.2, were excluded from the dataset. To respect the prompting 'context length' of LLMs, in instances where the prompt exceeded a length of 3000 words, a further refinement process was employed. This process involved the elimination of words with a tf-idf Sammut & Webb [2010] score below 0.2 and truncating the prompt to a maximum of 3000 words.

It is also important to note that the dataset does not encompass all calendar dates within the specified time range. This limitation emanates from the trading calendar days, and absence of relevant financial news headlines for certain dates.

Label ($\boldsymbol{x_r}$): The label is determined by the percentage change in closing prices from one day to the next, as defined in equation 9. This percentage change is categorized into three labels: {Rise, Fall, Neutral}, based on the thresholds specified in equation 10.

$$PCT_{\text{change}} = \left(\frac{\text{Closing Price}_t - \text{Closing Price}_{t-1}}{\text{Closing Price}_{t-1}} \right) \times 100\% \tag{9}$$

$$x_r = \begin{cases} \text{Fall} & \text{if } PCT_{\text{change}} < -0.5\% \\ \text{Neutral} & \text{if } -0.5\% \leq PCT_{\text{change}} \leq 0.5\% \\ \text{Rise} & \text{if } PCT_{\text{change}} > 0.5\% \end{cases} \tag{10}$$

D.2 BIGDATA22

D.3 IMDB REVIEWS DATASET

Table 10: Cluster Information and Differences for Different Runs

Cluster Number	Points	Cluster Split (%) Rise	Neutral	Fall	Entropy	Weighted IG
Base Data						
All prompts	317	23.66	59.62	16.72	1.3683	-
LLama-2-7b						
Cluster 0	175	25.71	54.29	20.00	1.4467	-0.0439
Cluster 1	46	21.74	65.22	13.04	1.2641	0.0152
Outliers	96	20.83	66.67	12.50	1.2364	0.0402
					Total Information Gain	**0.0117**
LLaMA-2-7b-chat + SFT						
Cluster 0	53	32.08	45.28	22.64	1.5289	-0.0269
Cluster 1	5	40.00	40.00	20.00	1.5219	-0.0024
Cluster 2	24	25.00	50.00	25.00	1.5000	-0.0100
Cluster 3	16	31.25	56.25	12.50	1.3663	0.0010
Outliers	153	25.49	62.09	12.42	1.3033	0.0314
Cluster 4	4	50.00	50.00	0.00	1.0000	0.0046
Cluster 5	5	60.00	40.00	0.00	0.9710	0.0063
Cluster 6	3	33.33	66.67	0.00	0.9183	0.0042
Cluster 7	3	33.33	66.67	0.00	0.9183	0.0042
Cluster 8	5	20.00	80.00	0.00	0.7219	0.0102
Cluster 9	4	0.00	100.00	0.00	0.0000	0.0173
Cluster 10	4	0.00	100.00	0.00	0.0000	0.0173
					Total Information Gain	**0.0835**
LLaMA-2-7b-chat + SFT + RLMF						
Cluster 0	7	28.57	42.86	28.57	1.5567	-0.0417
Cluster 1	43	23.26	55.81	20.93	1.4312	-0.0086
Cluster 2	44	20.45	56.82	22.73	1.4175	-0.0064
Cluster 3	17	35.29	47.06	17.65	1.4837	-0.0062
Cluster 4	23	30.43	52.17	17.39	1.4509	-0.0060
Cluster 5	26	26.92	57.69	15.38	1.3829	-0.0012
Cluster 6	27	25.93	59.26	14.81	1.3604	0.0007
Cluster 7	13	15.38	61.54	23.08	1.3347	0.0014
Cluster 8	10	40.00	50.00	10.00	1.3610	0.0023
Outliers	34	29.41	58.82	11.76	1.3328	0.0037
Cluster 9	4	50.00	0.00	50.00	1.0000	0.0046
Cluster 10	7	14.29	71.43	14.29	1.1488	0.0049
Cluster 11	17	11.76	64.71	23.53	1.2608	0.0058
Cluster 12	5	60.00	40.00	0.00	0.9710	0.0063
Cluster 13	10	10.00	70.00	20.00	1.1568	0.0067
Cluster 14	4	25.00	75.00	0.00	0.8113	0.0070
Cluster 15	4	0.00	100.00	0.00	0.0000	0.0173
Cluster 16	7	0.00	100.00	0.00	0.0000	0.0302
Cluster 17	15	6.67	93.33	0.00	0.3534	0.0480
					Total Information Gain	**0.1039**

Table 12: Statistics and breakdown of splits sizes

Category	Statistics
Number of data points	2111
Number of Rise/Fall/Neutral label	558 / 433 / 1122
Train/Test/Evaluation split	1477 / 317 / 317

Table 13: Date Ranges of news headlines in splits

Split	Num. Samples	Date range
Train	1477	2010-01-06 to 2017-06-27
Valid	317	2017-06-28 to 2019-02-12
Test	317	2019-02-13 to 2020-09-21

Table 14: Summary of the dataset columns with their respective descriptions.

Column Name	**Description**
Date	Date of the trading session
Opening Price	Stock's opening market price
Daily High	Highest trading price of the day
Daily Low	Lowest trading price of the day
Closing Price	Stock's closing market price
Adjusted Closing Price	Closing price adjusted for splits and dividends
Volume	Total shares traded during the day
Percentage Change	Day-over-day percentage change in closing price
MACD	Momentum indicator showing the relationship between two moving averages
Bollinger Upper Band	Upper boundary of the Bollinger Bands, set at two standard deviations above the average
Bollinger Lower Band	Lower boundary, set at two standard deviations below the average
30-Day RSI	Momentum oscillator measuring speed and change of price movements
30-Day CCI	Indicator identifying cyclical trends over 30 days
30-Day DX	Indicates the strength of price trends over 30 days
30-Day SMA	Average closing price over the past 30 days
60-Day SMA	Average closing price over the past 60 days