CONTRASIM: CONTRASTIVE SIMILARITY SPACE LEARNING FOR FINANCIAL MARKET PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce the *Contrastive Similarity Space* (ContraSim) paradigm that is able to form global semantic understanding between how daily financial headlines can affect market movement. ContraSim consists of two steps. 1) Weighted Headline Augmentation: We propose a method of augmenting financial headlines to create new headlines with known semantic distances to the original. 2) Weighted-Self Supervised Contrastive Learning (WSSCL): An extension of classical binary contrastive learning algorithms, WSSCL leverages the known distances between anchor and augmented prompts to generate finely grained embedding space that optimizes for similar news to be clumped together. We measure how well ContraSim is able to learn global financial information by parsing whether or not it inherently groups newslines of homogeneous market movement directions together, using a novel information density metric Info-kNN. We find that incorporating features from ContraSim into financial forecasting tasks has a 7% increase in classification accuracy. Additionally, we highlight that ContraSim can be used to find historic news-days that most resemble pertinent financial headlines of the day to help analysts to make better decisions for predicting market movement.

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. Predictably, with increased knowledge representation algorithms, researchers have tried to use these algorithms as a way to build better financial forecasting models to see if it is possible to “beat the market”.

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 financial news, to repeatedly perform well in market movement prediction. However, 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.

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]. Although, we observe an increased performance in market movement prediction from the inclusion of

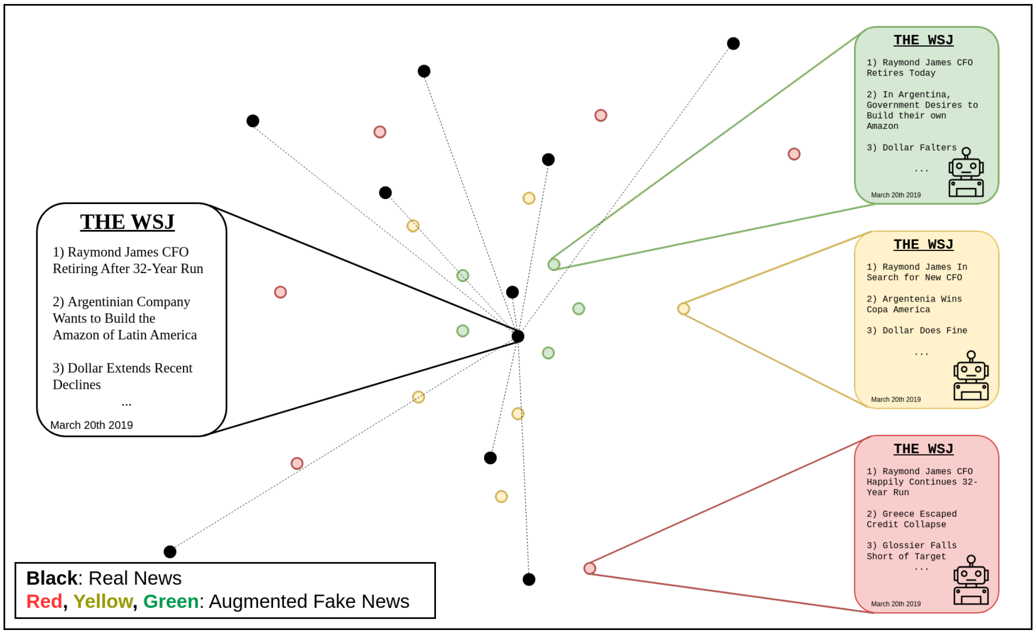

Figure 1: Contrasim generates 20 augmentations of the original March 20th 2019 Wall Street Journal The Wall Street Journal [2024] headlines. ContraSim generates these ablations with known distances from the original. We then use those ablations in a weighted self-supervised learning approach to generate a rich embedding space that pushes newslines of similar semantic meaning closer together. As a result, we can then later measure the distance between all other real news sources to see which historic newsline is most semantically similar to our newsline of interest.

pertinent textual information, state of the art (SOTA) language model techniques destroy the rich knowledge within financial news by predicting either a binary market direction prediction (eg. *Rise*, *Neutral*, *Fall*), or a regressive market movement percentage (eg. 5%, -1%). We are not only interested in predicting the direction of today's market, but also to **measure which days are most similar to the market conditions of today**.

In this work, we explore the domain of daily news headlines from the Wall Street Journal (WSJ) The Wall Street Journal [2024], as a method to link complex and noisy global knowledge to direction of the stock market movement, while maintaining the richness of information present within each day's headlines. For example, a WSJ headline: "Canada opens new oil pipeline to United States", has an effect on market conditions that may affect market movement. Using information from that headline alone may prove to be useful for a financial analyst, however if we are also able to extend the analysis by providing historic newslines from the rich corpus available to us, we can use that to make better decisions.

To achieve the rich embedding space we introduce a novel weighted self-supervised contrastive learning (WSSCL) approach that groups news-lines containing semantically similar headline information within a closer local proximity, than newslines (set of all the headlines on a day) of extremely disparate meaning. We introduce an augmentation system for newsline prompts that use LLMs to create modified prompts containing semantically identical or augmented newslines. Then using a WSSCL approach, we cluster these augmented prompts either closer or farther apart depending on the augmentation applied.

Additionally, to measure the degree of the semantic richness of the clustering algorithm, we introduce info-kNN, a modification of a k-Nearest Neighbours algorithm that uses concepts from information theory to better gauge the level of information clustering within our similarity space.

Contributions Our main contributions with this work are:

1. **ContraSim for Financial Headlines**: We propose the *Contrastive Similarity Space Embedding Algorithm* (ContraSim) a method for generating prompt augmentations with knowledgeable and rich similarity coefficients. In this paper we will show:

   a) Newsline similarity spaces generated by ContraSim allow for inter-day financial lookup, so financial forecasters can see which historic market days the current day is similar to.

   b) ContraSim learns a mapping between newslines and the direction of the market in an unsupervised learning fashion. This is achieved by showing that as an embedding similarity space is learned, structures emerge that increase global information on stock movement. *ie. By learning which prompts are most similar, we learn why stocks move*.

   c) Similarity embedding spaces created by ContraSim can be used in tandem with financial forecasting classification algorithms to increase task performance.

2. **Information Gain from Entropy of k-Nearest Neighbours (Info-kNN)**: We also introduce a method for evaluating the clumping of labelled embedding in a similiarty space with Info-kNN. Info-kNN, is an information theoretic approach to k-Nearest Neighbours that is agnostic to imbalanced labeled classes, and allows us to measure the level of information density that is created through our WSSCL paradigmn.

2 Related Works

Machine Learning in Financial Forecasting Early machine learning approaches for predicting movement in the stock market were based on applying classical statistical models to stock market data. Seminal model, Autoregressive Integrated Moving Average (ARIMA) Box & Jenkins [1970] used statistical time series models to predict movement direction. Following that, classical statistical models using a plethora of techniques from *Generalized Autoregressive Conditional Heteroskedasticity (GARCH)* Bollerslev [1986], *Vector Autoregression (VAR)* Sims [1980], and *Holt-Winters exponential smoothing* Holt [1957], and others Engle & Granger [1987], Kalman [1960], Hamilton [1989] were employed to capture more complex relationships in financial time series.

However, with classical statistical models, the financial modalities are typically confined to the use of tabular datasets. With the advent and explosion of Large Language Models (LLMs), financial models were better able to parse nonlinear relationships between market data and market direction. Additionally, the use of LLMs allows researchers to introduce more complex modalities into their models. For examples, market movement prediction accuracy has been increased by adding news articles Yang et al. [2020], sentiment analysis Yang et al. [2020], social media data Bollen et al. [2011], and more complicated financial earning calls Tsai & Wang [2016], to their models.

Contrastive Learning Contrastive learning has emerged as a powerful paradigm in unsupervised and self-supervised learning, leveraging the idea of learning through comparison. The fundamental objective of contrastive learning is to bring representations of similar data points closer while pushing representations of dissimilar data points further apart. One of the earliest methods in this area was SimCLR Chen et al. [2020], which used data augmentations and a contrastive loss to learn representations without labels. This approach was further refined by MoCo He et al. [2020], which introduced a memory bank to store negative examples, increasing the model's efficiency in handling larger datasets.

More recent advancements such as SimSiam Chen & He [2021] have shown that competitive representations can be learned without negative pairs, further improving efficiency and reducing computational requirements, making it more accessible for large-scale datasets commonly found in financial applications.

3 METHODS

In this section, we introduce ContraSim, a self-supervised contrastive learning algorithm that creates augmented prompts of varying degrees of semantic richness, and uses a weighted self-supervised learning paradigm to create a similarity space, with prompts organized locally via distance. Additionally, we 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 prompt similiarty.

3.1 CONTRASIM: CONTRASTIVE SIMILARITY SPACE EMBEDDING ALGORITHM

We formulate the steps of how a rich embedding space if created that pools days of similar stock movement together. The goal the described algorithm is to create an embedding space that puts market days with similar headlines together. With a dynamic space we can then look towards future financial days and collate which other days are most similar. Furthermore, we can investigate how strong of a predictive mechanism a similarity based embedding space is at predicting market movement, as compared to other predictive algorithms.

Let T be the training set that consists of all newslines $N_1, N_2, \ldots, N_k$, defined as:

$$T = \{N_i \mid 10 \leq |N_i| \leq 30,\ i = 1, 2, \ldots, k\} \tag{1}$$

$$N_i = \{h_{i1}, h_{i2}, \ldots, h_{in_i}\} \tag{2}$$

Where h_{ij} represents the j-th headline in the i-th newsline N_i, $n_i = |N_i|$ is the number of headlines in N_i, and the number of headlines satisfies the constraint $10 \leq n_i \leq 30$. Each newsline contains only the headlines from a specific day. Note that if a newsline contained more than 30 headlines, we randomly selected 30 from the set to reduce newsline complexity and computational demands.

For this experiment, we only use headlines, and omit any other financial tools to make better predictions. The goal of this experiment is to evaluate the augmentation techniques and how they are able to generate domain knowledge solely from comparing newslines with augmented pairs. For the purpose of simplicity we keep only the newslines and we leave the work on incorporating tabular data to further research.

1. Creating Headline Augmentations : For each remaining headline for each news day, we used LLaMA-3-7b-chat AI [2024] to generate 5 "reworded" headlines, $\hat{h}_{re}$, 5 "slight-ablated" prompts, $\hat{h}_{ab}$, and 5 "negative" prompts, $\hat{h}_{ne}$. To generate multiple responses from the same prompt, we employed a top-p random sampling technique. We sampled from top-p sampling of p = 0.9, temperature = 0.8, and a repetition penalty = 1.2. This setup enabled for responses that were randomly augmented, but still aligned to the details in the instruction prompt. Each headline augmentation is generated with the instruction shown in Table 1.

$$N_i^{\text{aug}} = \left\{\{h_{ij}^{(k,m)} \mid k = 1, 2, 3\}, m = 1, 2, \ldots, 5\right\} preservesemanticsimilarity \tag{3}$$

Here, $h_{ij}^{(k,m)}$ refers to the m-th variation of the k-th augmentation type (reworded, ablated, or negative) for headline h_{ij} in newsline N_i. The intuition behind the math here is simple. Each headline has 5 reworded, 5 slight-ablated, and 5 negative augmentations.

	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
	Slight Ablation 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 1: 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.

2. Generating Newsline Buckets: The goal of creating newsline weighted augmentations is to take the original unaugmented anchor prompt, and to create a list of 20 variations that are very either clearly semantically similar or clearly semantically different. We create a copy of each anchor newsline, shuffle the order of each headline, and from the top down we can perform 5 different actions: Take the a re-worded headline (**R**). Take a slight-ablated headline (**A**). Take a negative headline (**N**). Take a random headline from a different random day (**O**). Delete the headline (**D**). The augmentation action probability distribution of performing each action is a hyperparameter to be optimized. Some of these actions are indicative of positive, neutral, or negative similarly and we will use that concept later to generate similarity scores.

An issue that appears with this generating newsline ablations with a distribution of our 5 proposed actions, is that without modification, the mean similarity score for each augmentation is equal, and can have a low standard deviation. For training, we want our projector network to see encodings with both very high (near 1.0) similarity scores and very low (near 0.0) similarity scores. We want to avoid, our projector model to learn to put all prompts at a consistent distance of mean similarity score.

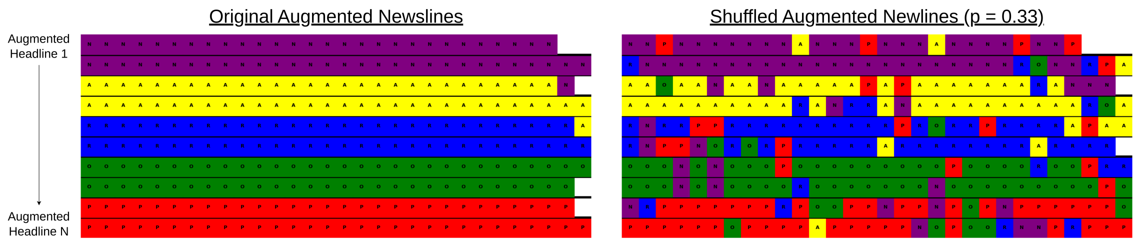

Figure 2: Each of the N newsline ablation is generated from first creating a long string of available actions. In this example, the global distribution of all augmentation actions (**R**, **A**, **N**, **O**, and **D**) are equally likely at 20%. To maintain a global distribution equal, while creating inter-newsline distributional variance we first organize augmentation actions sequentially, and then we randomly chose actions to be flipped according to probability p = 0.33, and flip action types according to the global distribution. These newsline buckets are then used to generate fully augmented newslines for training.

Our solution, is to create a tile permutation effect as described in figure 2. For each anchor newsline, we generate N = 20 augmented newslines. For each augmented newsline we first initialize bucket sizes. The distribution of bucket sizes is equal to the distribution found in the NIFTY dataset. We then sequentially fill buckets, B_i, with actions equal to the augmentation action probability distribution (AAPD) hyperparameter, such that all actions are done in order. Next, we iterate through each bucket and with probability p = 0.3, we randomly flip actions to another random action according to AAPD. We use these buckets to generate augmented newsline prompts.

The result of this flipping strategy is that we are able to create a series of augmented newslines such that the global distribution of actions types remains AAPD, but for each individual augmented newsline, we can observe varying degrees of similarity.

3. Creating Newsline Augmentations Using our generated newsline buckets, we next able to create full newsline augmentations. The strategy for generating each augmented newsline from our buckets is outlined in Algorithm 1.

4. Generating Similarity Scores In the process of creating newsline weighted augmentations, we associate a similarity coefficient to each action. Action **R**, that preserves semantic similarity has a similarity coefficient of 1.0. Actions **A** and **D**, that slightly modify the meaning of the original newline have a coefficient of 0.5. Finally, actions **O** and **R** have similarity coefficients of 0.0.

As described in equation 4, by taking the mean of all performed actions we produce a bounded similarity score between [0, 1]. Similarity scores will be used for training a projector network to create a rich similarity space that preserves newsline semantic closeness. We optimize our projector network to minimize the distance of highly similar newlines, and to maximize distances for newlines of low similarity scores.

Algorithm 1 Create Newsline from String

Require: N_i: Current news data row, N_i^{aug}: Dataset for augmentation, B_i: Augmentation bucket vector, H: Vector of all dataset headlines.
Ensure: List of processed news headlines
1: $newsline\_list \leftarrow []$
2: $headlines \leftarrow$ keys of $N_i^{\text{news\_list}}$
3: **for** $c \in B_i$ **do**
4:     **if** $c =$ **O then** Append a random headline from H
5:     **else if** $c =$ **R then** Remove and append a random rephrased headline from N_i^{aug}
6:     **else if** $c =$ **A then** Remove and append a random ablation headline from N_i^{aug}
7:     **else if** $c =$ **N then** Remove and append a random negative headline from N_i^{aug}
8:     **else if** $c =$ **D then** Remove a random headline from $headlines$
9: **end for**
10: **return** $newsline\_list$

$$\text{Similarity Score} = \frac{1}{|B|} \sum_{a \in B} S(a) \tag{4}$$

Where, B is a vector of augmentation actions, and $S(a)$ maps a single augmentation action to its similarity score such that:

$$S(\mathbf{R}) = 1.0, \quad S(\mathbf{A}) = 0.5, \quad S(\mathbf{D}) = 0.5, \quad S(\mathbf{N}) = 0.0, \quad S(\mathbf{O}) = 0.0$$

5. Weighted Self-Supervised Contrastive Learning (WSSCL) Now that we have generated augmented newslines from training set of anchor headlines, and we have given similarity scores to each of these anchor-augmentation newslines, we can proceed to generating our newsline 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 *Fall*) from the NIFTY dataset Raeid et al. [2024]. Additional details of SFT implementation are available from Saqur [2024]. Newslines 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 newslines only.

The optimization task we define for our projection network are defined by two novel loss functions: Mean Squared Error (Equation 5), and Continuously Weighted Contrastive Loss (Equation 6).

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}\left(max(S(\mathbf{p}_i, \mathbf{q}_i), 0) - s_i\right)^2 \tag{5}$$

$$\mathcal{L}_{\text{WCL}} = -\frac{1}{n}\sum_{i=1}^{n} s_i \cdot \log\left(\frac{\exp(S(\mathbf{p}_i, \mathbf{q}_i)/\tau)}{\sum_{j=1}^{n}\exp(S(\mathbf{p}_i, \mathbf{q}_j)/\tau)}\right) \tag{6}$$

$$\mathcal{L}_{\text{combined}} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{WCL}} \tag{7}$$

Where: $S(\mathbf{p}_i, \mathbf{q}_i)$ is the cosine similarity between anchor and augmented projections, s_i is the similarity score between the anchor and augmented projection, τ is the temperature scaling parameter, and n is the number of samples.

Both losses are variations on classical loss functions used in contrastive learning tasks, but are extended past binary classification of inter-prompt. $\mathcal{L}_{\text{WCL}}$, is a method introduced by Srinivasa et al. [2023], and extends supervised contrastive loss with a similarity weight term s_i, which incentivizes the model to ensure low distances between similar pairs, in a smooth continuous manner.

Initially, we employed supervised fine-tuning (SFT) on a LLaMA-3-8b-chat model using next sentence completion to predict "Rise", "Fall", or "Neutral". This was achieved by training only the LoRA layers. Subsequently, to train the projection layer, we froze all model layers except the last five and applied the contrastive learning approach described above. Finally, once the projection space was established, we trained the classification networks while keeping all other layers frozen.

3.2 Evaluating Similarity Space Information Richness

To evaluate the performance of our embedding model and the quality of the resulting embedding space, we employ several metrics that quantify how well the embeddings cluster data points according to their categories (*Rise*, *Fall*, *Neutral*). These metrics include our novel approach, Information Gain via Entropy of k-Nearest Neighbors (Info-kNN), as well as established metrics such as Nearest Neighbor Accuracy, Kullback-Leibler (KL) Divergence, and Jensen-Shannon Divergence (JSD).

1. Information Gain via Entropy of k-Nearest Neighbors (Info-kNN) We introduce a novel metric, Information Gain via Entropy of k-Nearest Neighbors (Info-kNN), which quantifies the clustering tendency of the embedding space by measuring the reduction in entropy of category labels among the k-nearest neighbors compared to a random distribution. This metric provides an intuitive interpretation of clustering effectiveness in terms of information theory, offering a new perspective on embedding evaluation.

For each data point i, we perform the following computations:

$$P_i(c) = \frac{\text{Number of neighbors with label } c}{k} \tag{8}$$

$$H_i = -\sum_{c=1}^{C} P_i(c) \log_2 P_i(c) \tag{9}$$

$$\overline{H} = \frac{1}{N} \sum_{i=1}^{N} H_i \tag{10}$$

$$H_{\text{max}} = \log_2 C \tag{11}$$

$$\text{IG} = H_{\text{max}} - \overline{H} \tag{12}$$

Here, $P_i(c)$ represents the local label distribution for category c among the k-nearest neighbors of data point i, H_i is the entropy of this distribution, $\overline{H}$ is the mean entropy across all data points, and H_{max} is the maximum possible entropy for a uniform distribution over C categories. The information gain measures how much the embedding space reduces uncertainty in category labels among neighboring points compared to a random distribution. A higher information gain indicates that the model effectively clusters similar data points, thereby enhancing the discriminative power of the embedding space.

Info-kNN extends k-Nearest Neighbours by being agnostic of label imbalances. In the NIFTY dataset used the ratio of rising, neutral, and falling, market days is 23%, 60%, and 17% respectively. Since Info-kNN measures the information gain associated with being in proximity to local points, over the total global distribution, we do not have inflated accuracy scores.

2. Additional Metrics In addition to our novel Info-kNN metric, we employ several established metrics to evaluate the quality of the embedding space. **1) Nearest Neighbor Accuracy** assesses the proportion of data points whose closest neighbor shares the same category label, providing a direct measure of clustering performance. **2) 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. **3) 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.

4 Results and Interpretations

Table 2 shows that a conjunction of projection, and LLM embeddings are better able to classify newslines 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.

Table 3 displays embedding space density metrics for a baseline, and our similarity space projection. We observe an increase in clustering accuracies in Info-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.

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

Metric	Baseline	$Class_{Proj}$	$Class_{LLM}$	$Class_{Both}$
Accuracy	.3333	.3434	.3522	**.3774**
F1 Score	.3333	.3389	.3833	**.4670**

Table 2: Accuracies and F1 scores for classification models, $Class_{Proj}$, $Class_{LLM}$, and $Class_{Both}$. Normally, NIFTY has a $Rise$, $Neutral$, $Fall$ split of (23%, 60%, 17%), we subsetted NIFTY to achieve a (33%, 33%, 33%) split.

Model	Info-KNN (k = 5)	KNN (k = 5)	KL-Divergence	JSD
Estimated Baseline	.5916	.4668	**.3539**	**.1054**
Similarity Space Projection	**.6248**	**.5142**	.3894	.1152

Table 3: Comparison of Baseline and Projection models across different evaluation methods: Info-KNN, KNN, KL-Divergence, and JSD. Note that finding true baseline values for these metrics on an unbalanced set of labels is nontrivial, and out of scope for this paper. As a result, estimated baseline values are a mean of 1000 cases of randomly distributed points following the (23%, 60%, 17%) label split.

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.

4.1 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.

4.2 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. During the initial sweep the augmentation action probability distribution (AAPD) was split at 20% each. Once we found a usable set of LR (0.01) and gamma values (0.85), we next performed a random sweep of 100 random configurations of the AADP, converging on an optimal value of around **O**: 0.4, **R**: 0.35, **A**: 0.125, **N**: 0.125, **D**: 0.05. Lastly a full hyper parameter sweep was performed again on learning rate, gamma, p, temperature and batch size.

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 Info-KNN (k=5) scores.

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.

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.

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

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`.

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.

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.

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`.

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

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

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`.

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.

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.

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.

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