Final Report - Topic Directionality in Financial Statements

Hanqin Zhou(hz2699) Xuan He(xh2465) Weiwei Jiang(wj2312)
Boquan Sun(bs3232) Jingyuan Wang(jw4000)

May 13, 2022

Abstract

Different from general sentiment analysis, topic directionality analysis task is to interpret the topic change direction (increase/neural/decrease) in a financial statement. Specifically, we are interested in the financial related topic like inflation, credit and growth. In this project, we have developed a financial statement topic directionality analysis procedure for PIMCO that is able to detect the direction for a given topic in a sentence. A supervised topic classifier model and a novel proposed distance model (position based and semantic based) are combined in our analysis procedure. The quantitative analysis on test data and the result of backtesting demonstrate that our proposed model reaches remarkable performance.

1 Introduction

The Topic Directionality Capstone project is sponsored by PIMCO, an American investment management firm. The intention of this Capstone is to build a model to parse financial statements and assess the directionality of various topics. For example, in the statement from US Federal Reserve Board, we analyze what topics are covered in each sentence. And for a certain topic assigned to the sentence, like inflation, we need to determine whether they are talking about it increasing or decreasing. More specifically, the scope of our interested topics is as follows: credit, fed funds rate, financial markets, geopolitical uncertainty, growth, housing, inflation, labor market, liquidity measures and quantitative easing.

One basic intuition behind this task is that financial statements typically include significant information about the market and the changes which have happened or are going to happen. If we can successfully calculate the topic directionality of financial documents, these extra information could be utilized as our guidance of our trading strategy. So once the topic directionality of each sentence has been gotten, we will aggregate the direction prediction for different sentences by timestamps and perform a backtesting based on the directionality.

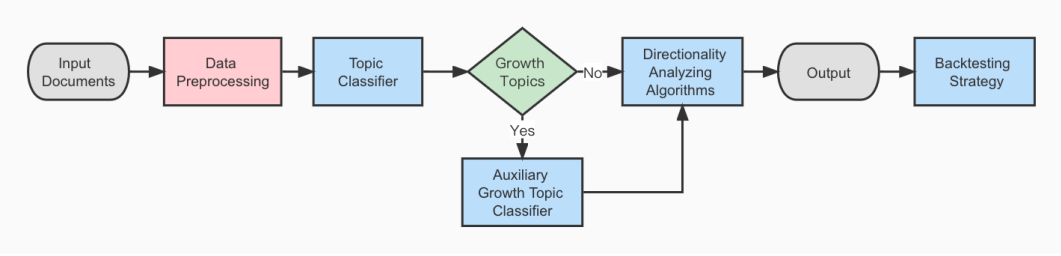


Figure 1: Proposed Model for Topic Directionality

To tackle this problem, we propose our Topic Directionality model as in Fig 1. For a given input document, text preprocessing techniques are first applied. Then, classification algorithm like Random Forest [1] is used to classify sentences into the interested topics based on the prediction probability. After that, an auxiliary growth topic classifier will be trained to distinct the growth topic sentences from non-topic sentences. For the direction analyzing algorithm part, we first manually tagged 1000 lines of the unlabeled data and then developed a three staged distance-based model with a direction

word dictionary in order to improve the interpretability and the transparency of the whole progress. In this model we recognized the featured topic words first, and find the nearest directional word describing the topic to make prediction of the sentence's potential direction. One thing need to be noted is that directionality is different from sentiment we discussed the most in other topics. The following table is a simple example to show the difference.

Sentence	Sentiment	Direction
"However, incoming economic data have tended to confirm that greater uncertainty, in part attributable to heightened geopolitical risks, is currently inhibiting spending, production, and employment."	negative	upward (geopolitical uncertainty)

Table 1: Sentiment and Direction inconsistency example

Finally, we build a portfolio strategy based on the signal from the directionality change and backtest it with the historical data of general financial index.

2 Data Cleaning and Exploratory Analysis

Since we are dealing with dirty text data, data cleaning steps are important. Below shows our cleaning pipeline for the text data:

Step 1. Replace symbols such as "\n", "\t", ";", "b\'" by space. "b\'" is a meaningless symbol observed in the beginning of the documents in our data.

Step 2. Remove website links in the text that start with "https://".

Step 3. Delete pure digit text.

Step 4. Delete sentences that are end-notes of the article. These sentences start with "Return to text" and "See [year]".

Step 5. Remove the digit followed after each sentence without space. They are sequence numbers of different paragraphs.

Step 6. Remove digit from the text since we don't need digit for our project purpose.

Step 7. Set text to lower case and extract tokens.

Step 8. Remove stopwords and lemmatize.

Step 9. Replace underlines by space.

Step 10. Remove single "s" and recover lemmatized "le" to "less".

The text cleaning steps are not a standard proprocessing pipeline for all text data. They are based on the observation of the data and some steps are created or modified to suit our data. For example, "less" is an important directionality word. Therefore we resume it after lemmatizing for the convenience of future directionality analysis. Also, note that step 1 to 10 should be conducted in a sequential order because the result of some step might effect other steps.

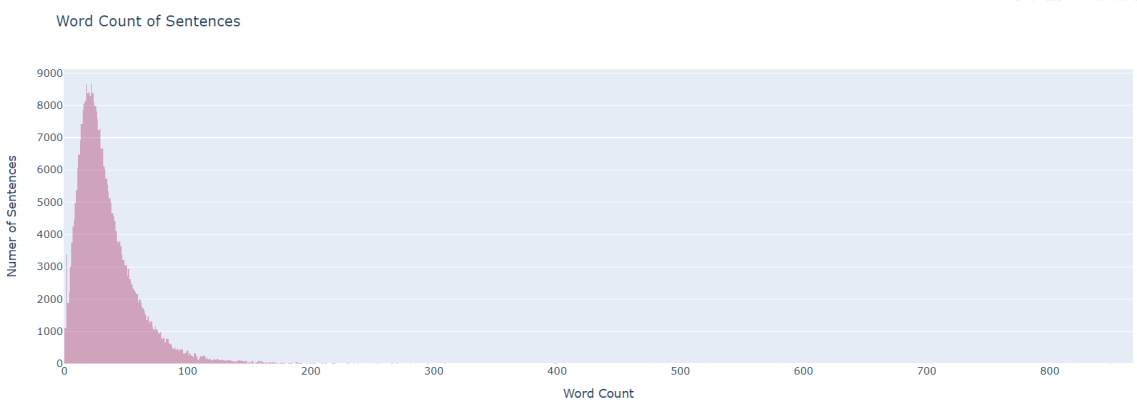


Figure 2: Frequency graph of the word counts of sentences in unlabeled data.

```
df_long_sentence = df_test_data[df_test_data['word_count']>100]
len(df_long_sentence)
```

```
9245
```

```
df_long_sentence = df_test_data[df_test_data['word_count']>100]
len(df_long_sentence)
```

```
102
```

Figure 3: Number of long sentences before (left) and after (right) the sentence split.

In the unlabeled data, we discovered anomaly in the distribution of number of words in each sentence. See figure 2 and 3. 9245 sentences have word counts more than 100. This is caused by the failure of separating sentences from the data source due to some special formats. To handle this issue, step 1 and 5 clean up the unreadable format in the original data and we also add up a sentence split operation between step 5 and 6. After splitting, the number of long sentences is reduced to 102, which could be ignored in the following discussion.

2.1 Labeled data

Labeled data contains roughly 400 financial sentences and their corresponding topics, in which one sentence is labeled with one or two financial topics. There are 13 topics among the sentences , which are credit,fed funds rate, financial conditions, financial markets, geopolitical uncertainty, global growth, growth, housing, inflation, labor market, liquidity measures, monetary policy and quantitative easing. see figure 4. After visualizing the distribution of the topics, we find it is imbalance, such as "growth" and "monetary policy" sentences are much more than others while "financial markets", "geopolitical uncertainty" and "global growth" are below average obviously. This phenomenon indicates that we need introduce methods to handle the imbalanced data when classifying topics.

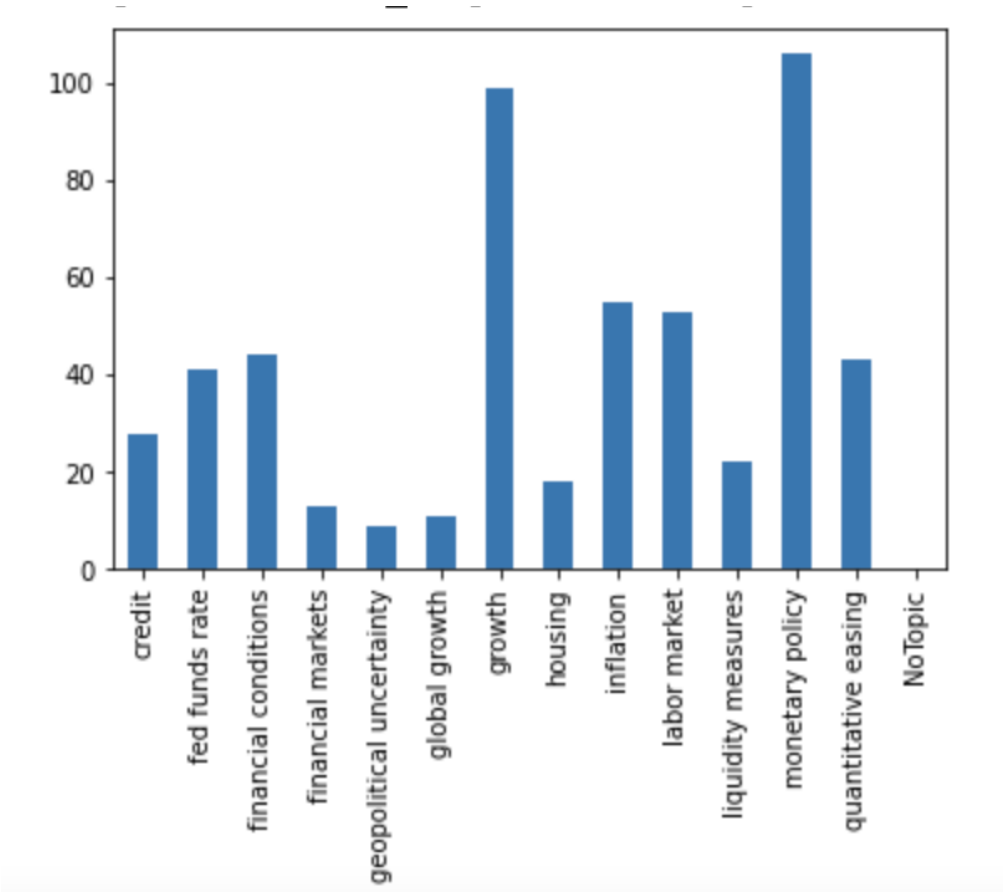


Figure 4: 13 financial topics distribution

As you can see from figure 5, A visualization of the 15 most common words across all labeled data are all about economy, market and policy due to the preprocessing of the primary data, including lemmatization and the removal of stop-words, punctuation and any other unnecessary characters. As a result, 'market' and 'inflation' are the most-talked about aspect in the labeled data. And there are other words are also common and related to our 13 topics like 'federal', 'price', 'rate', 'labor', etc.

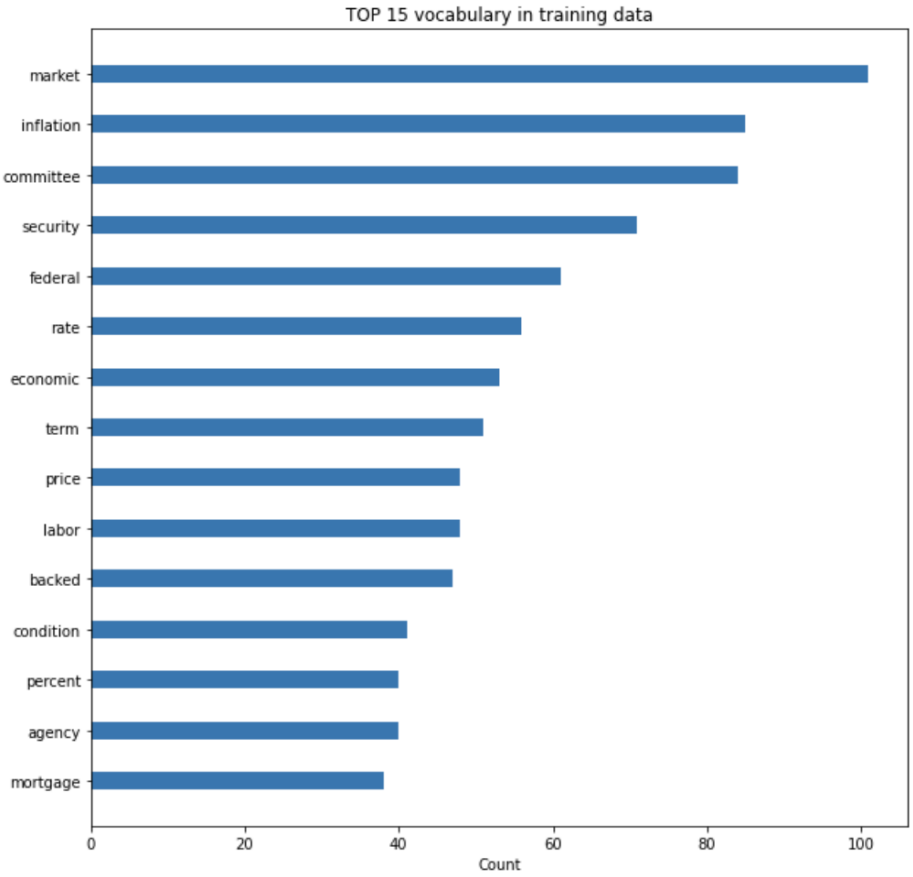


Figure 5: top 15 words in labeled data

2.2 Unlabeled data

The unlabeled data contains of 340,000 sentences extracted from tons of financial statements from US Fed and US central bank since 1997. As figure 6 shows, top words are still financial policy topics such as 'policy', 'rate', 'financial', 'bank', 'market', etc. The most common words have a big similarity in labeled data and unlabeled data, proving through a proper model classifying test sentences can be feasible and consistent.

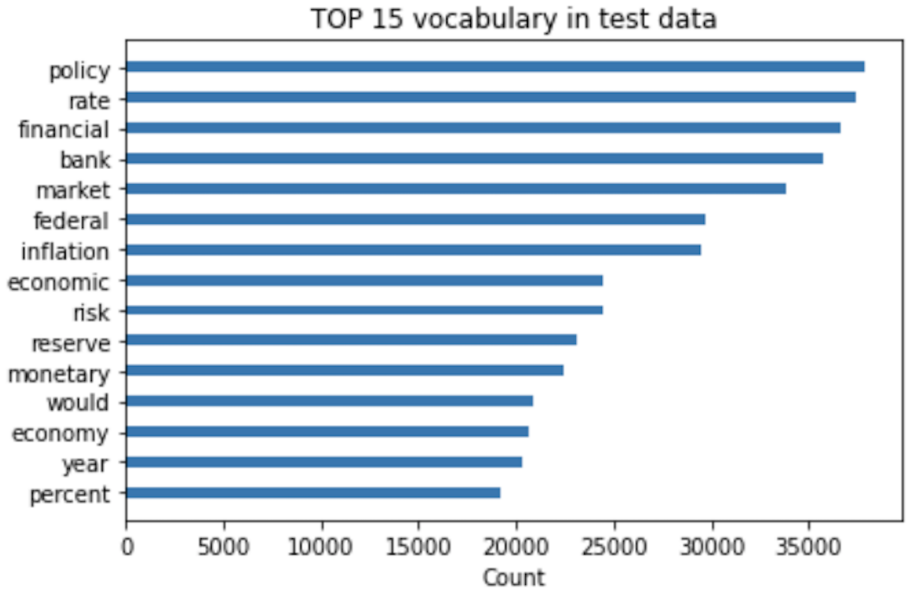
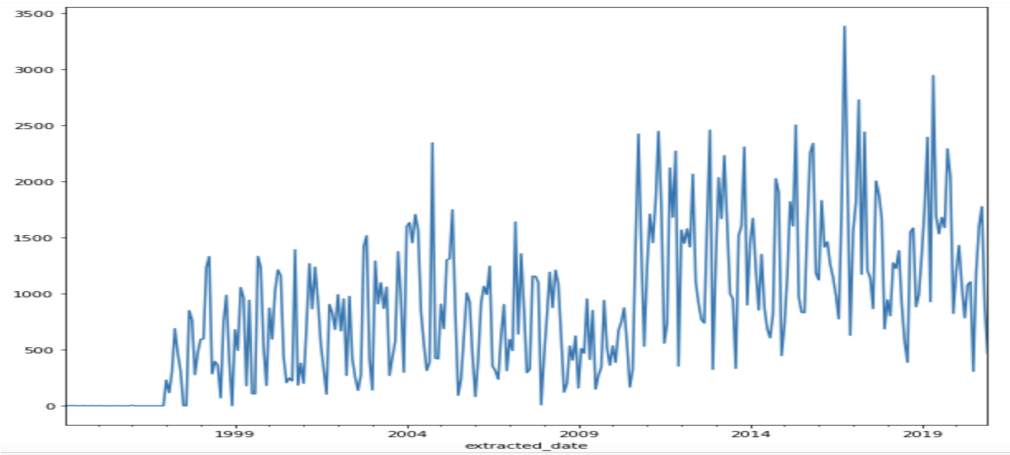
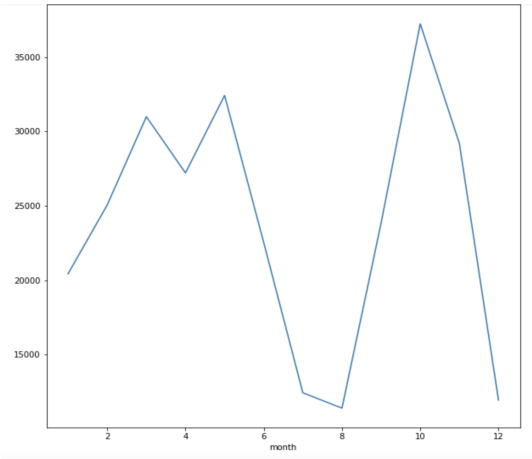


Figure 6: top 15 words in unlabeled data

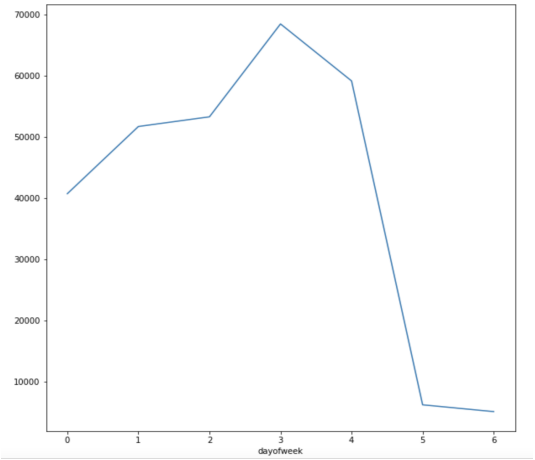
As shown in figure 7, We plot the sentence-time distribution from three aspects that are time series, monthly and day of week. The number of financial statements exhibits seasonality, while the overall trend is that from 2011 there are more financial sentences data than before. And sentences released in October are the most while less sentences stated in August. Also, financial statements are more likely to appear on Wednesday and Thursday rather than Friday and Saturday.



(a) time series of all sentences in unlabeled data



(b) count numbers of sentences for month



(c) count numbers of sentences for day of week

Figure 7: sentence-time distribution

3 Model Overview

All in all we have two phases in our model. In the first phase, our goal is to classify sentences into specific topics and in the second phase our goal is to predict the directionality mentioned in the sentence regarding specific topics.

1. **The First Phase - Topic Classification Model**
   (a) **Word Embedding:** Transform the text input into a format more agreeable for ML and NLP techniques
   (b) **Multi-Class Classification Algorithm:** Classify sentences in financial statements into the thirteen pre-determined topics
   (c) **Auxiliary Growth Topic Classifier:** Distinguish the true "growth" label sentences out of the predicted "growth" sentences of the Multi-Class Classifier.
2. **The Second Phase - Directionality Analyzing Algorithm**
   (a) **Supervised Learning Model**: Based on labeled data, fine tuned basic classification models.
   (b) **Unsupervised Learning Model**: Simulate the logic for human-beings to detect the direction in the sentence.
   First stage: Find out all the topic words which contributed the most when we are doing

topic classification.
Second stage: Find out all the potential directionality words in the sentence, which may talk about the direction of the sentence.
Third stage: Find out the (topic, directionality words) pairs which are closest to each others.
Fourth stage: Introducing a negative word gate. Calculate the direction index of the sentence based on the direction of the directionality word, the distance between directionality word and the topic word and whether there is a negative word ahead of the directionality word.

4 The First Phase - Topic Classification Model

4.1 Word Embedding

There are sever popular algorithms in Natural Language Processing to vectorize the input text format. Generally speaking, we can either using the pure frequency statistic method or the complexity language model to vectorize the input text. Specifically, FinBERT [2] is a open-source pre-trained BERT language model in the finance domain, which is handy in our case. In this Capstone project, we compared the performance of using TF-IDF vectorization [3] and the performance of using FinBERT embedding. Taking the performance and efficient into consideration, we identify the TF-IDF vectorization is the most suitable word embedding for this project.

4.2 Multi-Class Classification Model

4.2.1 Dataset preparation

In our project, one sentence may cover more than one topics. For example, the quoted sentence,'Overall financial conditions remain accommodative in part reflecting policy measures to support the economy and the flow of credit to US households and businesses', mentioned both 'financial conditions' and 'credit'. In order to further define a manual threshold to classify the topics, we use the following guidelines to process labeled data as suggested by our supervisor. If there are multiple labels for one sentence, for example, sentence 1 with labels A, B and C, we would build our labeled data as follows:

X	Y
Sentence 1	Label A
Sentence 1	Label B
Sentence 1	Label C

Table 2: labeled dataframe

Initially there are 389 sentences in our raw dataset. We firstly split the raw dataset into train set and validation set by ratio 3:1. To make sure each topic is distributed evenly in training and validation dataset, we stratified the whole dataset by 13 topics. After that, we split the multi-label sentences into separate sentences using the above techniques. Then we applied TF-IDF to transform the text data into matrices.

4.2.2 Model Training

Bear the small labeled dataset size in mind, it's not feasible to train complex models with lots of parameters. Therefore, we confined our potential models between Logistical Classification and Random Forest. After the dataset preparation step, we treat our problem as a typical multi-class classification problem and training our model on the train data. Then we use the model we trained to make predictions on the validation dataset based on the *predict_proba*(). Every topic will assigned with a probability and the total probability of thirteen topics will be summed up to one. We set our threshold at 0.25 after different experiments, which means that once the *predict_proba*() is over 0.25 for a topic, then we assign this topic as a label to that sentence.

4.2.3 Model Evaluation

We evaluated our model on validation dataset. Since our data is multi-label, we re-defined the measurement metrics by ourselves.

1. **Accuracy:** The predicted labels have to be exactly the same as the ground truth Both in quantity and content
2. **Precison:** The number of correct predicted labels divided by the number of predicted labels.
3. **Recall:** The number of correct predicted labels divided by the number of ground truth.
4. **F1 score:** Harmonic Average of Precision and Recall.

We evaluated our random forest model on TF-IDF and FinBERT embedding at different threshold. The result obtained after evaluation are shown in Table 2 and Table 3. We found that, generally speaking, TF-IDF performs better than FinBERT. After discussion, we figure out that this situation is caused by the feature of the financial terms. In FinBERT language model, the embedding tends to understand the meaning of a word, and more common the meaning is higher contribution when assigning the label. However, when we are focusing on the financial statement, the common meaning of a word should have a lower contribution than its financial meaning. For example, the main meaning of "housing" in our daily life is making some place your home, but in the financial statement, it refers to a particular aspect of the market. Therefore, we come to the conclusion that for this project, using TF-IDF may meet our need more.

And for the random forest model based on TF-IDF embedding, we found that under 0.25 threshold,the model performs the best. We reached our final result of accuracy 76.5%, precision 84.0%, recall 88.8% and 86%.

Threshold	Accuracy	Precision	Recall	F1
0.10	0.48	0.73	**0.97**	0.83
0.15	0.64	0.81	0.94	**0.87**
0.20	0.73	0.83	0.89	0.86
0.25	**0.76**	**0.84**	0.88	0.86
0.30	0.73	0.83	0.82	0.82
0.35	0.68	0.79	0.75	0.77
0.40	0.56	0.74	0.66	0.69
0.45	0.53	0.68	0.60	0.64

Table 3: Threshold Performance For RF on TF-IDF

Threshold	Accuracy	Precision	Recall	F1
0.10	0.22	0.51	**0.86**	0.64
0.15	0.37	0.61	0.80	0.69
0.20	0.55	**0.68**	0.76	**0.72**
0.25	**0.56**	0.63	0.66	0.65
0.30	0.52	0.61	0.66	0.60
0.35	0.46	0.51	0.50	0.50
0.40	0.36	0.40	0.38	0.39
0.45	0.27	0.32	0.29	0.30

Table 4: Threshold Performance for RF on FinBERT

4.2.4 Model interpretation

For further understanding of our Random Forest model, we plot the feature importance over each topic. In Random Forest model, feature importance denotes the decrease in node impurity, which can be used to reflect the most distinguish features of each class. Suggested by the result of our plotting , our Random Forest model demonstrates a certain semantic understanding ability and shows robustness

to a certain degree. For example, taking a glimpse at the feature importance of class 'financial market', the higher value features like 'financial','equity' and 'wealth' make sense in this specific class scenario.

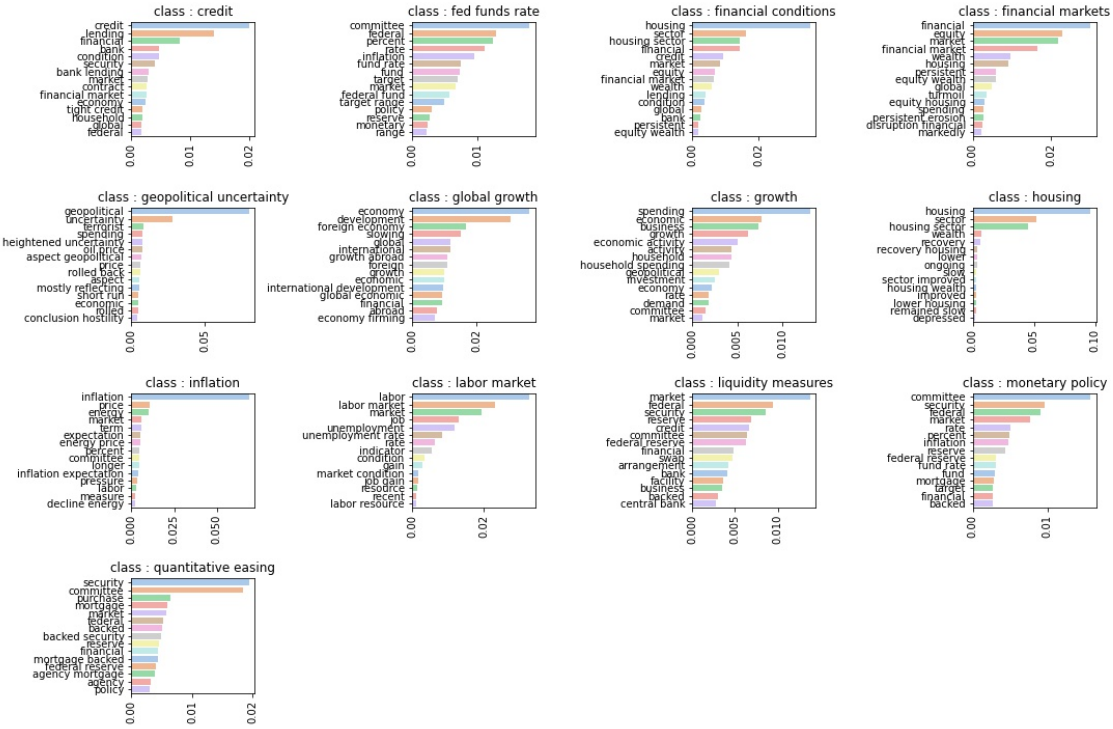


Figure 8: Top 10 feature importance for each topic.

4.3 Auxiliary Growth Topic Classifier

After training the model on train and validation data, we applied it to our unlabeled data to check its performance in a more general scenario. After prediction, for each topic, we randomly picked 20 sentence which is predicted to be in that topic to check the accuracy. We read this sentence manually and count the number of the miss labeled ones. The following is the result.

Topic	Miss Labeled Ratio	Labeled Count
quantitative easing	5%	1731
inflation	5%	26562
labor market	10%	26681
monetary police	25%	34907
credit	10%	10466
global growth	55%	3547
fed funds rate	0%	14281
growth	50%	246085
financial conditions	0%	6728
financial market	0%	3206
housing	15%	1703
geopolitical uncertainty	0%	226
liquidity measures	5%	779

Table 5: Miss Labeled ratio for each topic and the total assigned count

The model performs well for most topics. However, it performs particularly bad for "global growth" and "growth". For "global growth", bad performance may causes by the lack of the original data

points in labeled data. Since the sentence with the label "global growth" only takes up 1% of the total amount of the unlabeled data, this kind of miss labeling is acceptable. But, for "growth", the situation is different. The sentences labeled as "growth" take up more than 70% of the total data set. That makes it necessary to take a further path into this topic.

Firstly, we think it may caused by the general nature of the concept "growth". Unlike other topics which only mentioned some specific words indicating their topic like "geopolitical uncertainty", "growth" is really a general idea. Any sentence mentioned some comments about financial conditions, productivity, pricing can be considered as a sentence talk about growth. To avoid this kind of mislabeling, we build a second pass model to figure out whether a sentence is truly a "growth" sentence or a "non topic" sentence.

Secondly,we plot the univariate distributions using kernel density estimation for each class where y is the density and x is predicted probability. The result is as follows:

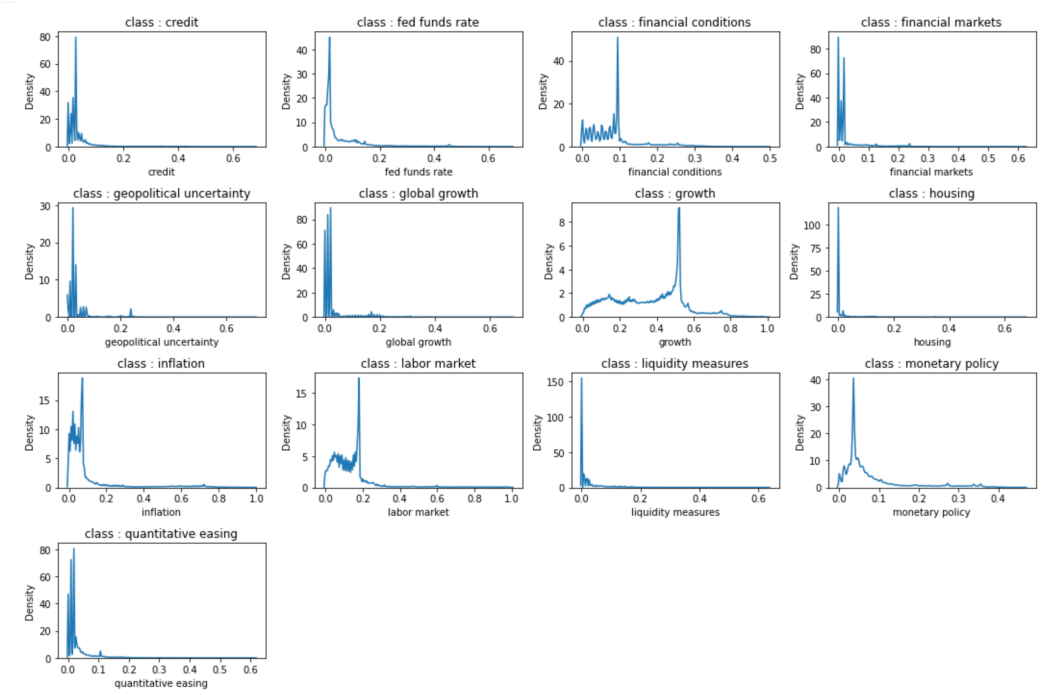


Figure 9: Density plot for predicted probability for all topics

From this plot, we can tell that there is a obvious high peak around 0.7 for "growth". And distribution of the probability is really even before this peak. However, for other topics, the probability mainly distributed from 0 to 0.4. It suggests us to increase the threshold for "growth" a little bit, to make the standard more strict for sentence to be labeled as "growth". This may help us balance the result and avoid miss labeling.

According to these two points, we come up with the following three approach to improve our model:

1. **Random Pick:** Randomly pick half of sentence with a label "growth" and remove the label.
2. **Higher Threshold:** Instead of setting a unified threshold for topics, set a higher threshold for "growth". According to the density plot, we tried 0.5, 0.55 and 0.6.
3. **Auxiliary Classifier:** Use another binary classifier to distinguish the true "growth" label sentences. The idea of this model is to figure out whether a sentence is truly a "growth". We labeled the train data by two labels, "is_growth" and "not_growth" and implement a logistic regression classifier on this label. In order to choose the threshold of the logistic classifier, we check the performance metrics of the model.

Threshold	Accuracy	Precision	Recall
0.30	0.92	0.88	0.81
0.35	0.91	0.76	0.86
0.40	0.91	0.72	0.90
0.45	0.92	0.68	1.00
0.50	0.88	0.56	1.00
0.55	0.86	0.48	1.00

Table 6: Threshold Performance for Logistic Classifier

As shown in the table, while the model with a threshold more 0.45 provide higher recall, the model with lower threshold provide higher accuracy. In our situation, it is hart to say which is better, since we also need to consider the total amount of "growth" label. For example, if accuracy is high but the total amount is large, it will include more miss labeled sentences. Hence we pick a set of these thresholds ([0.3,0.35,0.4,0.5]) and applied it to the overall test data. In this case, we can look at the metrics of the model together with its performance on the test set to see which one is better.

Result validation: For each path, we randomly sampled 20 lines from the sentences which has been assigned to the label "growth" and checked the whether it is truly belongs to "growth". At the same time, we collect the total amount of label "growth" among all the sentence to see the impact of the miss labeling.

Method	Miss Labeled Ratio	Proportion("growth"/total)
Original Random Forest	50%	72%
Random Pick	50%	34%
Higher Threshold(0.5)	45%	32%
Higher Threshold(0.55)	25%	12%
Higher Threshold(0.6)	20%	9%
Logistic Classifier(0.3)	14%	13%
Logistic Classifier(0.35)	20%	7%
Logistic Classifier(0.4)	10%	5%
Logistic Classifier(0.5)	35%	2%

Table 7: Growth topic classifer performance comparison

According to the result, the logistic classifier with a threshold of 0.3, provides the best balance between the miss labeled ratio and the label proportion. Although, in this model there are still some sentence like the a quote or a title of an article, eg. **"Longview: The Economic Outlook."**, are miss labeled into the class, we have limited the total amount of this situation into a acceptable range. As our next step is to recognize the directionality of the sentence, this kind of sentence will automatically be assigned as an neutral comment and their impact on our final model will be insignificant.
Another problem arose when we investigated the labeling the direction of each topic. We find sentences assigned as monetary policy or financial condition are similar with sentences with topics like quantitative easing and financial markets in term of semantic respectively. So we made an agreement with our mentors and decided to delete these two topics since they seem to be redundant topics providing no extra information. Another special case is the similarity between global growth and growth topic. They are semantically similar and the labeled growth sentences in our mentor provided data is far outnumber than the labeled global growth sentences. We combined the global growth and growth topic as one single growth topic as suggested by our mentors. The detailed modifications are shown in Table 8.

Topics Before Combined	Topics after Combined
global growth, growth	growth
financial condition/financial markets	financial markets
financial condition/housing	housing
financial condition/credit	credit
monetary policy/fed fund rate	fed fund rate
monetary policy/quantitative easing	quantitative easing
monetary policy/liquidity measures	liquidity measures

Table 8: Topic Modification Applied on the Training Set

5 The Second Phase - Directionally Analyzing Algorithm

5.1 Benchmark Model

5.1.1 Dataset preparation

Firstly we randomly sampled 100 sentences for each of the ten topics (credit, fed funds rate, financial markets, geopolitical uncertainty, growth, housing, inflation, labor market, liquidity measure and quantitative easing). Then based on the context of cleaned text , we labelled the true topic and direction for each sentence. The predicted topic might not be conformed to true topic based on our judgements since the initial topic predicting model might classify the sentence into a wrong topic. Overall, X should be the cleaned text for each sentence, y should be the direction labels that we manually assign each sentence to: increase (+1), neutral(0) and decrease(-1).

Overall initially there are 1000 sentences in our raw dataset (100 sentences for each topic) and we delete the sentences whose predicted topic is not conforming to the true topic based on our judgments. Since in real case, the topic of the sentence is not known in advance so we combined all of the topic together and train the overall sentences as a single classification model to get a general result. Then we evaluated our model on each topic to check the model performance on different topics. We firstly split the raw dataset into training dataset and test dataset by ratio 3:1. To make sure direction label is distributed evenly in training and validation dataset, we stratified the whole dataset by direction labels.

5.1.2 Model Training

Since it is a typical multi-class classification problem and bearing the small training dataset size in mind, we choose traditional simple supervised machine learning model including Logistic Regression, K Nearest Neighbor and Random Forest [1] as our baseline models. Firstly we trained and tuned the hyperparameters on the training dataset, then we fitted models with best parameters to make predictions on our validation dataset and get results.

5.1.3 Model Evaluation

We evaluated our model on validation dataset. Since our case is a typical multi-class problem, we used sklearn built-in metrics accuracy score, precision score, recall score, f1 score as our evaluation measurements. The results for our ten individual topic models and combined result are as below.

Topic	Accuracy	Precision	Recall	F1
Overall	0.61	0.58	0.52	0.52
financial markets	0.75	0.27	0.31	0.29
geopolitical uncertainty	0.65	0.39	0.45	0.41
growth	0.50	0.61	0.47	0.51
housing	0.68	0.71	0.68	0.63
inflation	0.52	0.29	0.35	0.30
labor market	0.50	0.34	0.43	0.38
quantitative easing	0.67	0.49	0.47	0.46
credit	0.68	0.34	0.41	0.37
fed funds rate	0.77	0.38	0.42	0.40
liquidity measures	0.58	0.39	0.42	0.38

Table 9: Logistic Regression Performance

Topic	Accuracy	Precision	Recall	F1
Overall	0.62	0.73	0.48	0.45
financial markets	0.75	0.27	0.31	0.29
geopolitical uncertainty	0.65	0.42	0.50	0.43
growth	0.58	0.69	0.53	0.56
housing	0.42	0.27	0.47	0.33
inflation	0.52	0.18	0.33	0.24
labor market	0.54	0.37	0.45	0.39
quantitative easing	0.57	0.61	0.61	0.57
credit	0.64	0.23	0.30	0.26
fed funds rate	0.80	0.27	0.33	0.30
liquidity measures	0.63	0.46	0.45	0.43

Table 10: K Nearest Neighborhood Performance on topics

Topic	Accuracy	Precision	Recall	F1
Overall	0.60	0.74	0.45	0.42
financial markets	0.75	0.27	0.31	0.29
geopolitical uncertainty	0.57	0.36	0.41	0.36
growth	0.33	0.27	0.31	0.23
housing	0.48	0.58	0.44	0.38
inflation	0.48	0.17	0.31	0.22
labor market	0.46	0.29	0.38	0.30
quantitative easing	0.52	0.72	0.62	0.50
credit	0.72	0.24	0.33	0.28
fed funds rate	0.80	0.27	0.33	0.30
liquidity measures	0.71	0.47	0.53	0.49

Table 11: Random Forest Performance on topics

The above are three baseline models for each ten topics individually and overall result. We found that the overall accuracy is 62% which is much better than than the random guess 33%. In particular, inflation,growth and labor market have low accuracy. By checking the confusion matrix, we found that the majority of the positive and negative labels are classified as neutral label. One of the possible explanation will be that the key words for distinguishing the directionality are ambiguous, model cannot extract and learn the direction words very well so they have low accuracy. It is also noteworthy that the recall rate and F1 score is very low. Low recall rate occurs when most of the positive values are never predicted. The low recall rate and F1 score occurs for two reasons. Firstly our sample is unbalanced with 340 number of neutral labelled sample, 210 number of negative labelled sample and 110 number of positive labelled sample. This imbalance can lead to a falsely perceived positive effect

of a model's accuracy, because the input data has bias towards one class, which results in the trained model to mimic that bias. Secondly our sample size is pretty small and model aligns too closely to a minimal set of data points, that caused over fitting problem. Thus the model learns more from majority class and cannot predict minority class well which is positive labelled sample in our case.

These are also the limitations for our benchmark models. Firstly, due to limited resource and constrained time, it is not possible for us to label more sentences which will result in small training dataset. Secondly, our benchmark work is a generalized model and is not specifically fit for our problem, we need a specific method designed to solve our case which leads to the distance models.

5.2 Distance based Model

Unlike the benchmark model, distance based model is an unsupervised model which takes unlabelled sentences and topic classification model as input and outputs the directionality of the sentences. The working principle of the model is: it finds the topic words in the sentence that indicate the topic classification results of the sentence. Then we use a self-built directionality dictionary to find the closest directionality words to the topic words. After getting the directionality words that tend to describe the direction of the underlying topic, aggregate the directions to denote the overall direction of the underlying topic in the sentences.

5.2.1 Directionality Dictionary

We construct a directionality dictionary based on the experience in the data labelling. The dictionary consists of about 150 words, each labelled as -1 or 1. The words indicate the direction of economic subjects. Typical words are for instance, increase and decrease, high and low. Note that we built this dictionary from our own because directionality is different from sentiment [4] and current open source word dictionaries are aiming to point out the sentiment instead of direction. While direction is the trend or movement, the later is focusing more on the feeling or subjective judgement.

5.2.2 Topic Word Recognition

Topic words are the one or more words in the sentences that strongly show the sentence belonging to that topic. For instance, in a cleaned sentence discussing labor market, "The August unemployment rate in Pennsylvania was percent, down points from a year ago and down from a peak of percent immediately following the recession", Word "unemployment" is the topic word of the sentence.

To get the topic words for each sentence, we use Tree Interpreter[5] to interpret the topic classification model and calculate the feature contribution of every word in the data. Then we extract the words with high contributions (with normalized contribution larger than 0.1). For sentences that do not have any word with contribution larger than 0.1, we extract the first four words with the highest contributions. Table 12 on next page is an example table showing the topic words and the contribution of them. We use a dictionary for each sentence to record the words and their contribution.

5.2.3 Direction Word Detection

1. **Stem Based Detection**

   There is subtle difference between our manufactured directionality dictionary and indicator words in financial sentences. For example, sometimes 'increases' and 'increase' can cause a bad case that they miss each other. Therefore, stemming[6] will be a powerful tool to handle the problem. Firstly we try our best to collect all kinds of direction words with different tenses. Secondly we chop off the ends of words including our direction words and all words in financial sentences to remove affixes. The final step is to pair the direction words with indicator words and record the pairing information. In a nutshell, stem based detection is mainly for the aim that any kind format of indicator words in sentences is supposed to pair with a direction word.

2. **Similarity Based Detection**

   As mentioned above, we manually designed a direction word dictionary for this project under the guidance of our mentors. The key represents a direction word, for example, recede, elevate,

Sentence	Topic words
(growth) There is no doubt that technical innovations such as the steam engine, railroads, electricity, and the automobile led to higher productivity growth, economic growth and, living standards.	[('growth', 0.24652699623739796), ('economic', 0.2272003648539155)]
(housing) The fall in home prices during the recession has given households a greater appreciation of the risks of leveraged investments in housing.	[('housing', 0.36336850290312495)]
(financial markets) Through both of these channels, the so-called wealth effect and the more general impact on consumer sentiment, equity valuations can and do have an impact on consumption and on macroeconomic performance.	[('equity', 0.26031935984468), ('wealth', 0.11579410437384399)]
(liquidity measures) One way we can help to support the availability of dollar funding is by engaging in currency swaps with other central banks.	[('swap', 0.07253924912078118), ('dollar', 0.055507390584155096), ('bank', 0.05203609931405033), ('central bank', 0.048948869369245016)]

Table 12: Topic words and their contributions of sampled sentences. The last sentence has four topic words because the highest contribution is less than 0.1 and model could not decide if it is adequate to indicate the topic.

boost. The value denotes the direction word is negative (-1) or positive (+1). Although we carefully examined 100 sample texts for each topic, it's still hard to design a complete direction word dictionary. Inspired by the concept of word vector [7], we decided to make a dictionary augmentation by computing the word similarity.

Despite the fact that our topic direction prediction model is an unsupervised model and it's hard to use any pre-trained model to perform transfer learning, the semantic space of words in our data set would be mostly consistent with other corpus. In NLP field, word vectorization is an methodology to project words or phrases from vocabulary to the corresponding vector space. Generally speaking, words and phrases that share the similar syntactic regularities and semantic regularities would have a similar vector projection. Based on these knowledge, we would use the cosine similarity to denote the semantic similarity of two words. Mathematically, it measures the cosine of the angle between two vectors projected in an N-dimensional vector space. The closer the vectors are in the N-dimensional vector space, the higher similarity is.

Specifically in our case, our dictionary augmentation works as follows. For a given input text, we would firstly split the sentence and iterate every token in this sentence. Taking the grammatical tense, singular and plural form into consideration, we apply stemming to make an unified format. Then for each token in the given input text, we use the pre-trained Spacy word vector [8] to compute the cosine similarity between the token and the defined direction words in our designed direction dictionary. We then compare the computed similarity score with our pre-defined similarity threshold to distinguish whether this token should be considered as a direction word. In our project, we conduct our following distance based prediction with two different threshold (0.6 and 0.8).

It is worth nothing that word vector not only considers the semantic regularities but also the syntactic regularities. In other words, sometimes, two different words that have a high cosine similarity could also be antonyms due to the syntactic regularities [9]. This feature introduces some undesired noise to our direction word detection, we should be aware of this disturbance in our following distance based model prediction.

3. **Position Distance Based Prediction**

After we detected the topic word and the direction word, we make a prediction based on find the closest pairs of these word. Although this method is not perfect, it is indeed a simple and feasible method worth trying.

For one sentence, suppose we detect m potential directional words S, which are similar to the direction word we found, and n topic words F, which contribute the most to its topic classification result.

$$S = [s_1, s_2, ...s_m]$$
$$DW = [dw_1, dw_2, ..., dw_m]$$
$$Sim = [sim_1, sim_2, ..., sim_m]$$
$$F = [f_1, f_2, ...f_n]$$
$$C = [c_,c_2, ...c_n]$$

For s_i in S, dw_i is the precollected directional word for this potential directional word. And sim_i is the similarity of these two words. When one potential directional word has more than one similar precollected directional word, we choose the one with the highest similarity.

For f_i in F, c_i is its contribution to the classification result, provided by tree interpretor.

First, we found the position of the words in the sentence.

For potential directional words: $Idx = [Idx_{s_1}, Idx_{s_2}, ..., Idx_{s_m}]$

For topic words: $P = [[p_1, p_2, ...]_{f_1}, [p_1, p_2, ...]_{f_2}, ...]$
Here, one potential directional words only has one index, but one topic word may has multiple postions. Because when we produce potential directional words we used its index to identify it, but when it comes to the features it is different.
Then there are two ways to find the closest word pair.

First, for each potential directional words we find an occurrence of feature, which is the closest to this word.

$$Pair(S, P) = [(s_1, p_{i_1, f_{j_1}, s_1}), (s_2, p_{i_2, f_{j_2}, s_2}), ..., (s_m, p_{i_m, f_{j_m}, s_m})]$$

Second, for each topic word we find the cloest direction word.

$$Pair(P, S) = [(p_1, s_{i_1, f_{j_1}, p_1}), (p_2, s_{i_2, f_{j_2}, p_2}), ..., (p_m, s_{i_m, f_{j_m}, p_m})]$$

Then we calculate the final direction index as following in four ways:

(a) Simple method: Simply calculate the average of directionality for all the pairs.

$$Dir = \frac{1}{cnt(Pairs)} \Sigma Dir_{P_i(s)}$$

(b) Distance weighted: Calculate the direction index as the weighted sum of the pairs direction according to the word distance within each pair.

$$Dir = \frac{1}{cnt(Dist(Pair))} \Sigma (\frac{1}{Dist(Pair)} * Dir_{P_i(s)})$$

(c) Contribute weighted: Calculate the direction index as the weighted sum of the pairs direction according to the topic word's contribution.

$$Dir = \frac{1}{\sum c}\Sigma(C_{P_i(p)} * Dir_{P_i(s)})$$

(d) Distance & Contribution weighted: Calculate the direction index as the weighted sum of the pairs direction according to the word distance within each pair and the topic word's contribution.

$$Dir = \frac{1}{(\sum c) * cnt(Dist(Pair))}\Sigma(C_{P_i(p)} * \frac{1}{Dist(Pair)} * Dir_{P_i(s)})$$

$Dir_{P_i(s)}$ refers to the directionality for a pair. It is calculated based on the word's direction and whether there is a negative gate word before the direction word. The negative gate word is a word like "not". And if there is a negative gate word before the direction word, the direction of the pair will multipled by -1. For stem method, it is the direction of the direction word(a integer in $0, -1, 1$). For similarity method, it is the direction word's direction multiple by its similarity with the pre-collected directional word.

Then for each sentence we get a direction index indicating its direction. And base on the index we set a reasonable threshold to decide whether its direction is 0,-1 or 1.

4. **Semantic Distance Based Prediction**
We also explore some semantic analysis methods rather than just simple position distance detection to obtain a more realistic and semantic distance for our pairs of direction words and topic words. In semantic distance based prediction, we are under the theory basis of dependent grammar which holds that there is a subject-subordinate relation between words, which is a kind of binary non-equivalent relation. In a sentence, if a word modifies another word, the modifier is called a dependent word, the modified word is called a head, and the grammatical relationship between the two is called dependency relation. We use SpaCy to implement the dependency analysis of sentences and draws the dependency diagram by Networkx. Firstly, we represent the dependency relationship by a tree graph, which is called a dependency parse tree. Dependency parse trees are used to express the dependency of words in a sentence.

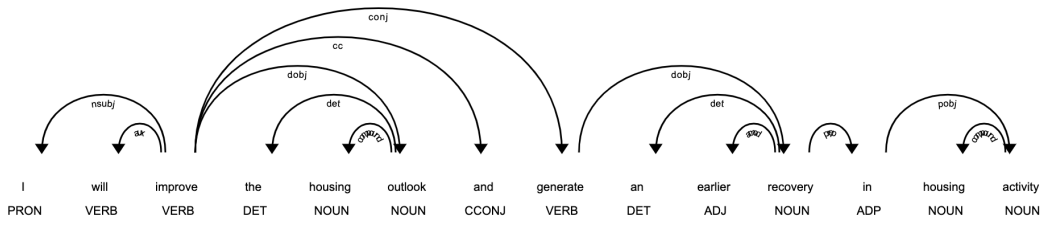


Figure 10: example of a dependency parse tree.

After we have the dependency parse tree by leveraging Spacy, we could also use networkx to build an undirected graph by the relation of dependency parse tree. Then we can find the shortest path of pairs of words we want to replace the original position distance.

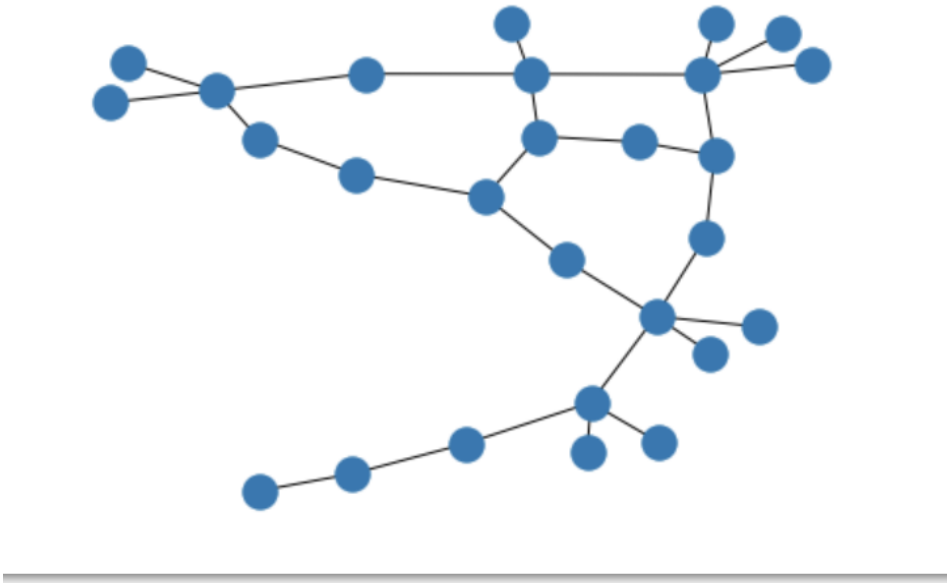


Figure 11: example of the undirected graph.

In this model, we build a dependency parse tree and an undirected graph, finding the shortest path as our semantic distance that explain our directionality model more precisely.

5.3 Result

5.3.1 distance based model

As mentioned above, we have stem based detection and similarity based detection to establish connection between our directionality dictionary and financial sentences indicator words. We evaluate two methods by comparing their accuracy, macro precision, macro recall and macro F1-score on 10 topics data. From the table below, stem based method have a remarkable better performance than similarity based detection. The reason is that our direction dictionary model is relatively complete for addressing most of problems while the similarity method can cause it ambiguous when the target words are synonym or antonym. Therefore, we deploy stem based method and continue to evaluate direction word based pairing and topic word based pairing.

Methods	Accuracy	Macro Precision	Macro Recall	Macro F1
similarity based detection	0.72	0.47	0.46	0.46
stem based detection	0.76	0.55	0.53	0.54

Table 13: Comparison of stem and similarity methods

Direction word based pairing means that firstly find the direction words and then find a closest topic word for every direction word, while topic word based pairing find the topic word first and then look for a closest direction word for every topic word. Intuitively, the two methods are not supposed to make a big difference on final results because they have advantages in respective way. And the table below also shows that they are almost of no difference. Therefore, we construct the distance based model with stem based detection and topic words based pairing for next step evaluation.

Methods	Accuracy	Macro Precision	Macro Recall	Macro F1
direction word based pair	0.76	0.55	0.53	0.54
topic word based pair	0.76	0.54	0.53	0.54

Table 14: Comparison of direction and topic word based pairs methods

This part we compare position distance based prediction and semantic distance based prediction. It reaches the conclusion that semantic analysis for sentences can improve performances when some words are distant from each other physically but close semantically. For example, house price will go as people wish, becoming higher and higher. "House" is distant from "higher" but we think "higher" exactly describes "House". Therefore, our distance based model will deploy by using the optimal methods which are stem, topic word based and semantic distance based.

Methods	Accuracy	Macro Precision	Macro Recall	Macro F1
position distance	0.76	0.54	0.53	0.54
semantic distance	0.77	0.56	0.55	0.56

Table 15: Comparison of direction and topic word based pairs methods

5.3.2 Confusion matrix

We use threshold 0.1 to classify our predictions as the final signals. Our distance based model performs well on our ten topics, we use confusion matrix to show the performance of our model. As you can see from 12, our overall accuracy is higher than benchmark model. And most sentences are classified correctly while only a very small portion(less than 10%) of predictions are opposite to the labels.

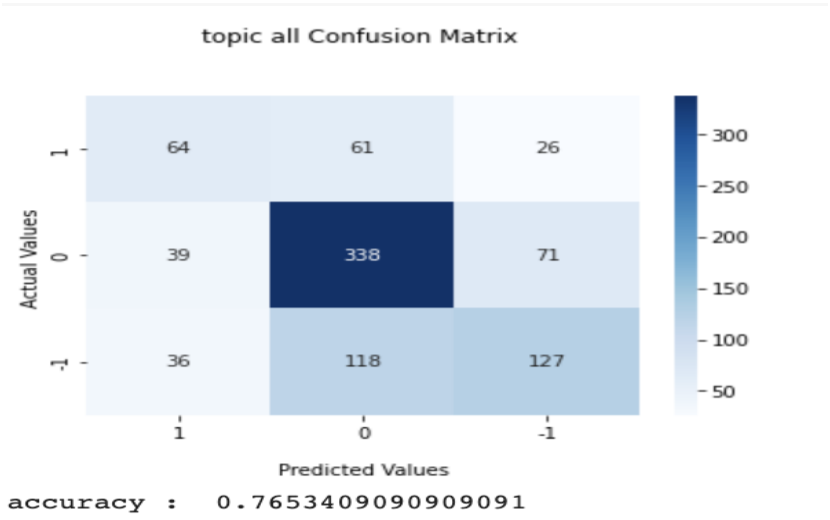


Figure 12: overall confusion matrix

There are 10 different topics among all aspects of financial environment, thus we need to evaluate our model on every topic. We selected representative topics shown as figure 13, it shows that the model do not have obvious deviation on different topics, but there indeed exists that financial market topic and fed funds rate topic have a larger accuracy than quantitative easing topic. Quantitative easing is a relatively special topic, which will talk about something about purchase or hold long-term security rather than things about increasing or decreasing. However, 'purchase' and 'hold' are also the feature words in our model, making our model not sensitive to this topic.

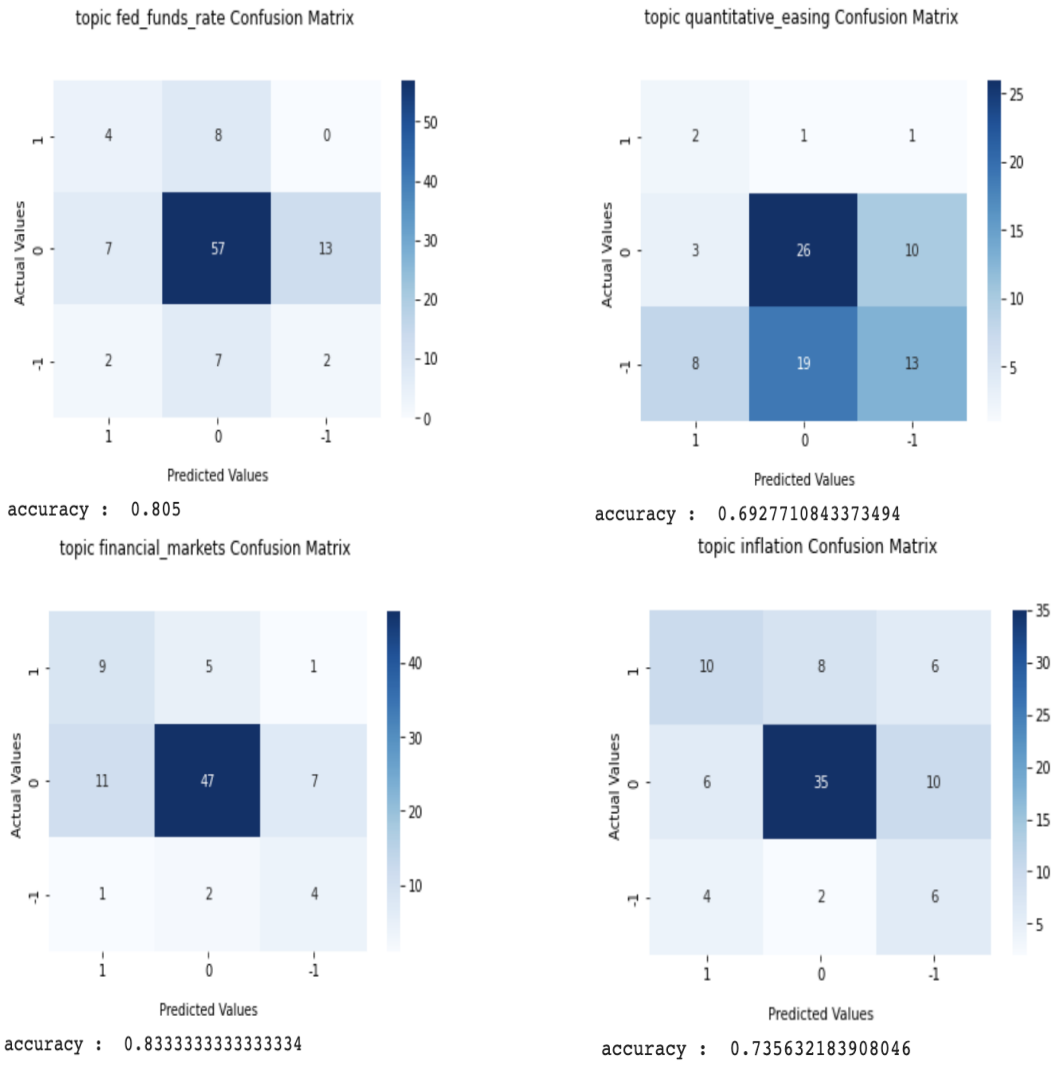


Figure 13: 2 topics confusion matrix

5.3.3 Comparison with benchmark

We evaluate benchmark work and distance based model together to see our improvement on the specific task. Because limited resource and constrained time, our training data for benchmark model is unbalanced and thus the models cannot balance class weights very well. The macro precision and macro recall are divergent indicating the loss of ability to predict the result in a balanced way. On the contrary, our distance based model specifically fit our problem and have a balanced performance on macro precision and macro recall. In the situation without enough labeled data, our unsupervised model reach a higher accuracy and better class balance meantime.

Methods	Accuracy	Macro Precision	Macro Recall	Macro F1	ROC AUC
Logistic Regression	0.61	0.58	0.52	0.52	**0.76**
K Nearest Neighborhood	0.62	0.73	0.48	0.45	0.74
Random Forest	0.60	0.74	0.45	0.42	0.73
Distance Based Model	**0.77**	0.56	**0.55**	**0.56**	0.67

Table 16: Comparison between benchmark model and our distance based model

6 Time Series Analysis and Backtesting

6.1 Time Series Analysis

The third stage following topic classification and directionality assessment is time series analysis, where we group the text by their extracted date and topic, and calculate the directionality of a certain topic in a certain date.

Before that, let's have an overview of the result of the directionality model. We have 105168 rows of data which is classified as single topic after dropping all the none topic and multi-topic data. Among the total 123677 rows that has at least one topic, 6194 rows are multi-topic sentence, which forms only a small portion. And also because our distance based model is not good at dealing with multi-topic sentences, we could simply discard them. And among 117483 rows of single topic sentence, their is 12315 duplicated rows, which results in 105168 rows.

Although having more than 100,000 data points, the distribution of each topic across the time is too sparse. For a given date and given topic, it is highly likely that their is no sentence or only one or two sentence mentioning that topic. To deal with this problem, we use two methods to draw the time series graph.

The first one is aggregating the directionality by month. Because of the sparsity of topic distribution and a relatively long time scale of the data, a lower granularity would make the graph more readable. We calculate mean directionality score (range from -1 to 1) for each month and topic. However the mean is not always reliable since in most dates, a given topic only have one or two sentences that are indicating its directionality. Thus we also draw the number of sentences related to that topic in every month. The sentences that indicated negative direction of the topic is separated from the positive sentences by different color in a stacked bar plot. Figure 14 and Figure 15 show the mean directionality and number of sentences in housing topic in a monthly manner.

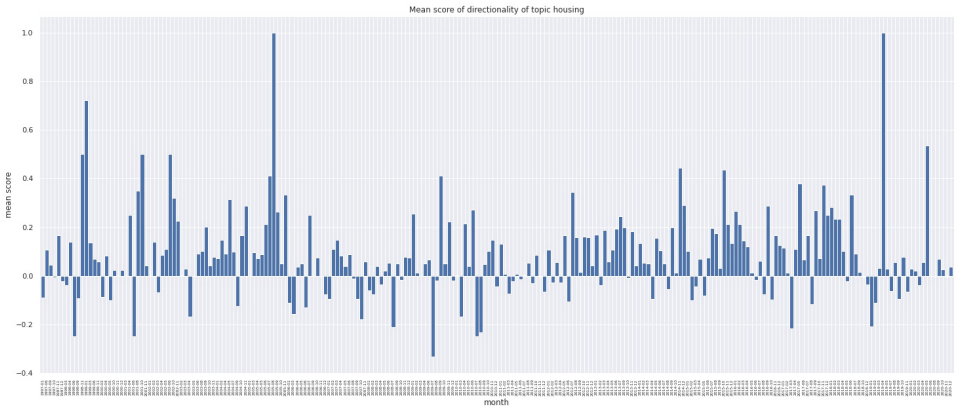


Figure 14: Mean directionality score of topic housing, aggregated by month. The score is ranging from -1 to 1.

The first method looks into the directionality in a monthly scale. One problem regarding this approach is that the contiguous dates are separated into different group. For example, 01/31/2019 and 02/01/2019 are two adjacent dates. But continuity of them is broken if we aggregate the data by month.

The second method is designed to handle the broken up of continuity. We apply a rolling window of length 100 (days) to the data and calculate the mean of the directionality score and sum of the number of sentences. Note that in this method there are a lot of dates that don't have a directionality score, the rolling window would ignore all the NAN values in the window and calculate only the non NAN

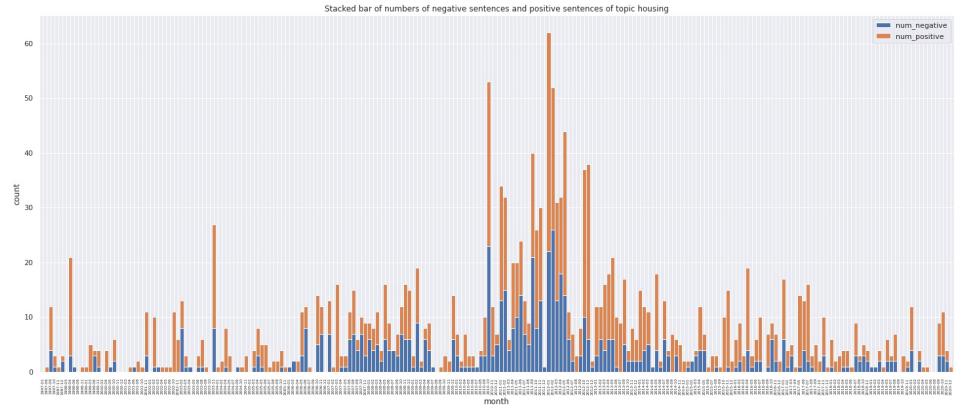


Figure 15: Number of negative and positive sentences of housing aggregated by month. The blue bar is the number of negative sentences while the orange bar is the number of positive sentences

values. Figure 16 and Figure 17 show the mean directionality and number of sentences in housing topic in 100 days rolling windows.

Figure 16: Mean directionality score of topic housing, rolling by 100 days windows. The score is ranging from -1 to 1.

6.2 Backtesting

The last part of our project is to explore the practice use our directionality predict model. We consulted 23 public financial indices (U.S. Government:Credit, U.S. Aggregate, S&P 500, U.S. Issued High Yield, Nasdaq 100, Russell 2000, U.S. Issued Investment Grade, KBW Bank, Mortgage-Backed, Bitcoin, KBW Bank ETF, MSCI Emerging Markets Index, CPI, Vanguard Real Estate Index Fund, FMCC, FNMA, GNMA, CPI Historical Data, Bloomberg Commodity, Crude Oil WTI Futures, Gold Futures, US Corn Futures, US Wheat Futures) to see if we can construct a portfolio strategy based on the market signal provided by the directionality of different topics extract from the statements.

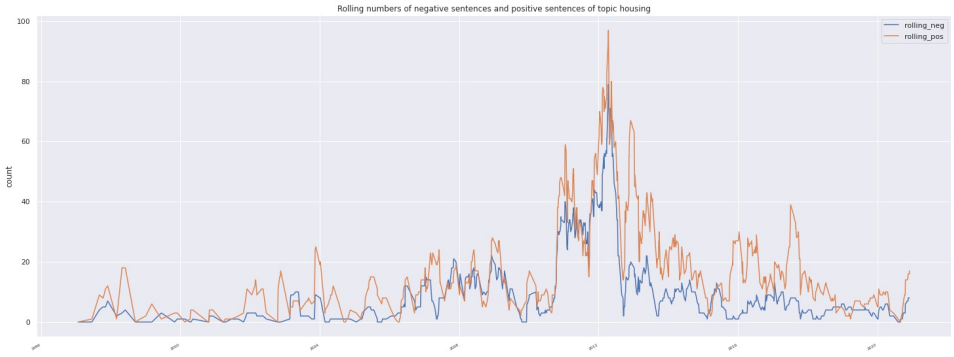


Figure 17: Number of negative and positive sentences of housing, rolling by 100 days windows. The blue line is the number of negative sentences while the orange line is the number of positive sentences

6.2.1 Portfolio Strategy

In the unlabeled data, we predicted the topic of each financial statement in the first phase model, and the directionality of the topic (-1, 0, or 1) during the second phase model. The ambition of our project is to exploit the directionality signals for financial investment. The directionality of a certain topic implies its trend, whether it's developing or declining, is the price increasing or decreasing? We assume that the trend of a certain topic would reflect on the price change of a certain index that relates to the topic. In this section, we would like to find some highly related topic-index pairs, where the directionality of the topic help us predict the price change of the corresponding index.

The strategy of our model is: at a certain trading day, if we receive a positive direction signal, we open one long position on the index and close it after a designated length of period; for negative direction signal, we open one short position and close it after the designated length of period. The holding period can be chosen as 1 for daily time horizon, 21 for monthly time horizon and 252 for yearly time horizon. Table 17 shows a toy example of how to trade based on the directionality signal in a monthly time horizon. Trade action only depends on the directionality signal and is independent of the previous tradings and positions.

date	price	directionality	action	return
05-16	100	1	open 1 long position	0
05-17	101	0	...	0
...	...	...	...	...
06-14	105	0	...	0
06-15	103	0	close 1 long position	0.03
06-17	104	0	...	0

Table 17: A toy example of the trading strategy in a monthly horizon, assuming the directionality from 05/17 to 06/15 remains 0.

To evaluate the strategy performance, we calculate the annual return, sharpe ratio and maximum drawdown. The return of the strategies in daily time horizon, i.e. holding period is 1 day, is calculated in cumulative manner through the whole time span. The return of the monthly time horizon (holding period is 21) and yearly time horizon (holding period is 252) is calculated in separate manner, that is, we calculate each trade action respectively and aggregate the performance of all the trade actions.

6.2.2 Result

We test 230 pairs of topics and indices in holding period of day, month and year. To choose which topic-index pair is profitable, we compare its performance with the benchmark, i.e. simple buy and

hold strategy on the same index.

Strategy returns for all pairs are calculated from six year ago to now. However, due to missing data, the strategy might not cover the whole time span.

When iterating through all the topic-index pairs, we pick out the pairs that have sharpe ratio larger than 1 and also larger than the benchmark sharpe ratio, the annual return of the directionality strategy should also exceed the benchmark annual return. Table 18 to **??** displays the performance of these pairs. We does not discover any well performed pair in yearly time horizon. This is reasonable because the effect of certain financial event will eventually fade away and it's unlike to last for a year.

topic	labor market
index	S&P U.S. Issued High Yield
annual return	4.77%
benchmark annual return	0.87%
sharpe ratio	1.4047
benchmark sharpe ratio	0.1386
max drawdown	-3.61%
benchmark max drawdown	-22.86%

Table 18: performance of topic-index pairs in daily time horizon

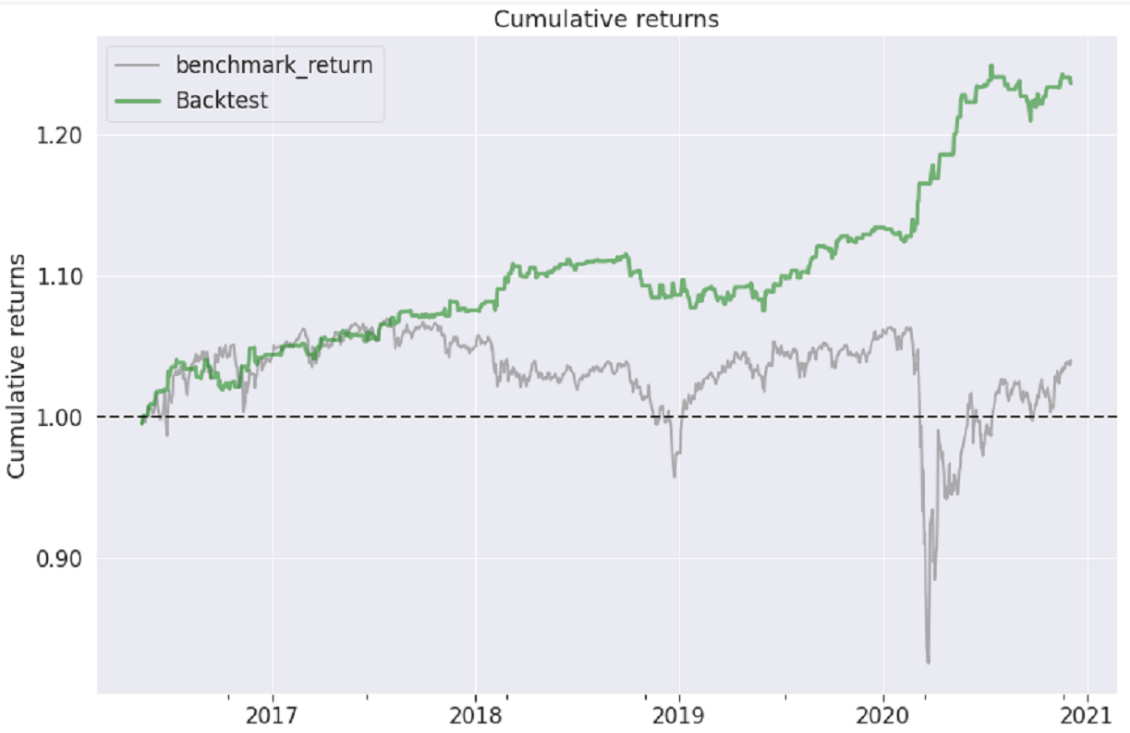
topic	labor market	housing	fed funds rate	labor market
index	S&P 500	S&P 500	Nasdaq 100	Nasdaq 100
annual return	29.12%	16.09%	34.91%	27.9%
benchmark annual return	13.64%	13.64%	15.2%	15.2%
sharpe ratio	1.0942	1.2472	1.3156	1.4944
benchmark sharpe ratio	0.7517	0.7517	0.7655	0.7655
max drawdown	-3.83%	-4.07%	-4.43%	-4.28%
benchmark max drawdown	-33.92%	-33.92%	-28.03%	-28.03%

Table 19: performance of four topic-index pairs in monthly time horizon

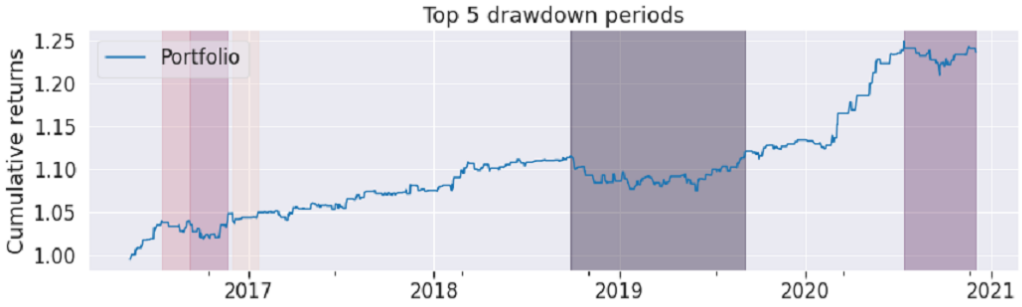
The best topic-index pair provides sharpe ratio around 1.5. The annual return of monthly trading strategy is much higher than daily strategy because we calculate the former for each trading action, when the trading period is small, the annual return tend to be a large number. While the latter are calculated cumulatively and the annual return will become smaller.

We also find that labor market topic directionality is a good price change indicator for many topics. The directionality strategy is more stable as it has lower max drawdown compared to benchmark.

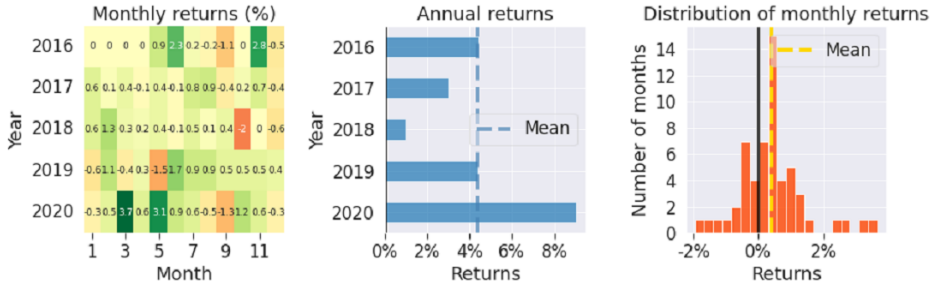
Figure 18 plots the performance of labor market-S&P U.S. Issued High Yield pair in daily time horizon. From the plot, we find that the directionality strategy performs especially well at large drawdown period of the index, meaning that the directionality signal predicted by our model is good at taking trading actions when price volatility is high. This feature makes sense because in high volatility period, the huge price change also comes with a strong directionality signal in financial statements, which is easy for our model to catch.



(a) Cumulative return of labor market - S&P U.S. Issued High Yield. Green line is the return directionality strategy. Gray line is the benchmark.



(b) Top 5 drawdown period of the directionality strategy of labor market - S&P U.S. Issued High Yield.



(c) Monthly and annual returns of the directionality strategy of labor market - S&P U.S. Issued High Yield.

Figure 18: Performance overview of the directionality strategy of labor market - S&P U.S. Issued High Yield.

7 Conclusion

In conclusion, we firstly developed a topic classifier model to classify financial sentences which were extracted from tons of financial statements from US Fed and US central bank since 1997 into multiple topics. Then we developed a distance model (position based and semantic based) which takes unlabelled sentences and topic classification model as input and outputs the directionality of each sentence. Our novel method can detect the direction of each financial sentence (increase/neutral/decrease) which is a distinct from the traditional sentimental analysis aimed for detecting the emotion of a sentence(positive/neutral/negative).

We validated our method both quantitatively and qualitatively. Firstly, in terms of quantitative comparison, our distanced based model outperforms the benchmark model in terms of accuracy, macro recall rate and F-1 score. Secondly, based on time series analysis and backtesting method, our model also has practical use in financial world. If we construct a portfolio strategy based on the market signal provided by the directionality change, the return of some financial index really outperforms the return of benchmark strategy(just hold stocks). Our qualitative and quantitative results are satisfactory in each step (data cleaning,model building and model validation) which is a complete and cohesive process.Thus, our unsupervised directionality model is validated both in model performance and real life practice use.

8 Future work

In spite of the success of our distanced model, there also exists some limitations. Thus, We propose some next steps for improvement in the future.

Firstly, our dataset is small and imbalanced. We can incorporate more pre-labelled financial documents in the dataset which might solve the overfitting problem. Secondly, for the distanced model based on semantic rule, our current strategy is based on dependency parse tree and find shortest path as distance. In the future, we may utilize deeper and more complicated semantic rules to construct our strategy. Thirdly, our current directionality dictionary is extracted from our manually labelled sentence based on our subjective judgement. Due to the limitation of resources and time, our current dictionary might be insufficient and incomplete, we might find some other ways to generalize our directionality dictionary to make it more accurate and complete.

References

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[3] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[4] Yasir Ali Solangi, Zulfiqar Ali Solangi, Samreen Aarain, Amna Abro, Ghulam Ali Mallah, and Asadullah Shah. Review on natural language processing (nlp) and its toolkits for opinion mining and sentiment analysis. In *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–4. IEEE, 2018.

[5] Pulkit Sharma, Shezan Rohinton Mirzan, Apurva Bhandari, Anish Pimpley, Abhiram Eswaran, Soundar Srinivasan, and Liqun Shao. Evaluating tree explanation methods for anomaly reasoning: A case study of shap treeexplainer and treeinterpreter. In *International Conference on Conceptual Modeling*, pages 35–45. Springer, 2020.

[6] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011.

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[8] Hendrik Heuer. Text comparison using word vector representations and dimensionality reduction. *arXiv preprint arXiv:1607.00534*, 2016.

[9] Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. *arXiv preprint arXiv:1605.07766*, 2016.