A Bridge from Federal Reserve Meeting Minutes to US Treasury Markets

NLP method for Financial Markets

Guanwen Cheng, Chunyi Li | S117 | 8/7/2020

Github Repository: https://github.com/aaroncgw/fednlp

Table of Contents

1. Projective Objective

1.1 Project Objective

The objective of this project is to apply NLP methods on texts published by Federal Reserve Open Market Committee (FOMC) to quantify the communications in a systematic manner and find the relationship between the text and Treasury yield curve.

1.2 Project Background

Financial Markets react too many types of information, such as GDP growth and unemployment rate. Traditionally, such quantitative factors can be easily incorporated into trading or investment models, which provide market practitioners execution signals. However, along with the rise of Deep Learning, qualitative factors play more and more important roles in the investment world. One main category among qualitative factors is language analysis based on NLP. A perfect example is the wording in documents and policies from government, especially regulators, such as Federal Reserve.

FOMC has eight regular meetings per year to determine U.S. monetary policy. The policy is based on current economy environment and forecasted situations. Meanwhile, the policy itself also impacts future economy. The endogenous relationship between policy and reality complicates their cause-effect relationship. After each FOMC meeting, Fed publishes press conference minutes, statements, as well as scripts in text, as a result, the market regularly observes significant volatility around FOMC meetings. However, because the complexity of real-world financial markets, it is not straightforward to quantify the exact impact from FOMC meeting on markets.

Note that, unlike certain NLP models, which may be able to achieve 90%+ prediction or classification accuracy, models predicting directions of financial market movements are extremely challenging, because what at stake is billions (if not trillions) of dollar. Usually, 60%-70% accuracy indicates good performance, while 70%+ accuracy may help one secure a job in hedge fund.

2. Data Overview

There are two components of the data. One is FOMC meeting minutes (text) and Treasury yield curve (numerical) for modeling. Second, Loughran and McDonald Sentiment Word Lists is for sentiment analysis.

Meeting minutes data was web scrapped from
https://www.federalreserve.gov/monetarypolicy/fomccalendars.htm
The year 2004 was chosen as the start-year given the acceleration of release dates to 3 weeks, and improved clarity in explanations of committee's decisions and views (Danker,2005)

Treasury yield curve is obtained from U.S. Department of The Treasury official site:
https://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yield

Loughran and McDonald Sentiment Word Lists
https://sraf.nd.edu/textual-analysis/resources/
Financial Sentiment Dictionaries, which are sentiment dictionaries for general and financial
sentiment analysis. It's used to look up sentiment score for each word.

We used several scripts to scrape and get data from these sources. They're under the data/utility folder.

3. Exploratory Analyses

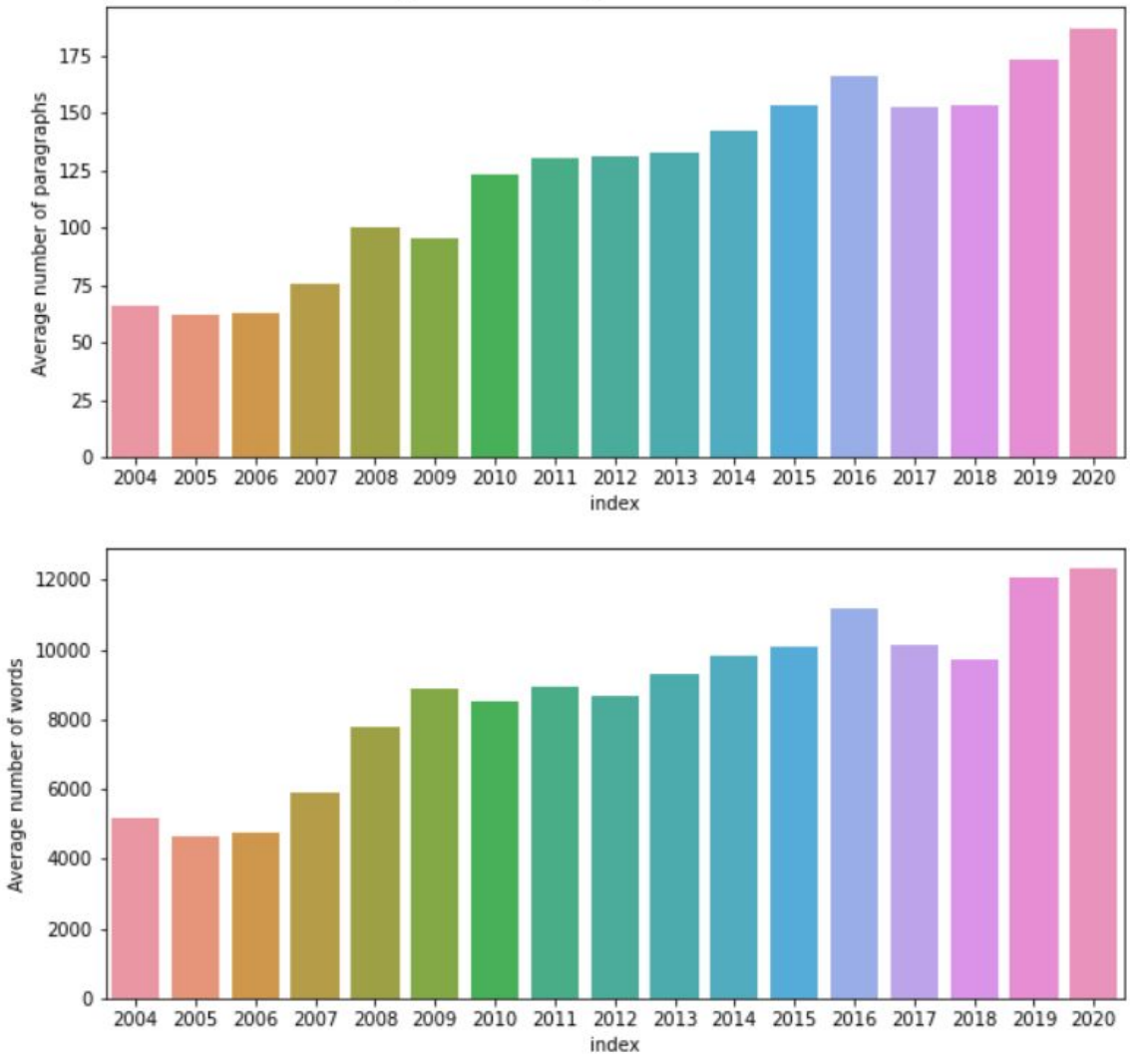
3.1 Meeting Minutes Summary

The meeting minutes have the following basic summaries:
Total number of files: 131
Number of paragraphs: 16006
Number of sentences: 41414
Number of words: 1126076

First file: 2004/20040128.txt
Last file: 2020/20200610.txt

From figure 1, we observe an acceleration in the numbers of paragraphs and words overtime. Before the Global Finance Crisis (GFC 2008), FOMC minutes were 60 ~ 75 paragraphs, and between 4500 ~ 6000 words. After the crisis, monetary policy became more important to US economy. FOMC members need to address more aspects on policy consequences. Since last year, the minutes have been extended even longer because of increased complexity from both the peaking of current economic cycle and massive internal trade conflicts.

Figure 1



3.2 Sections

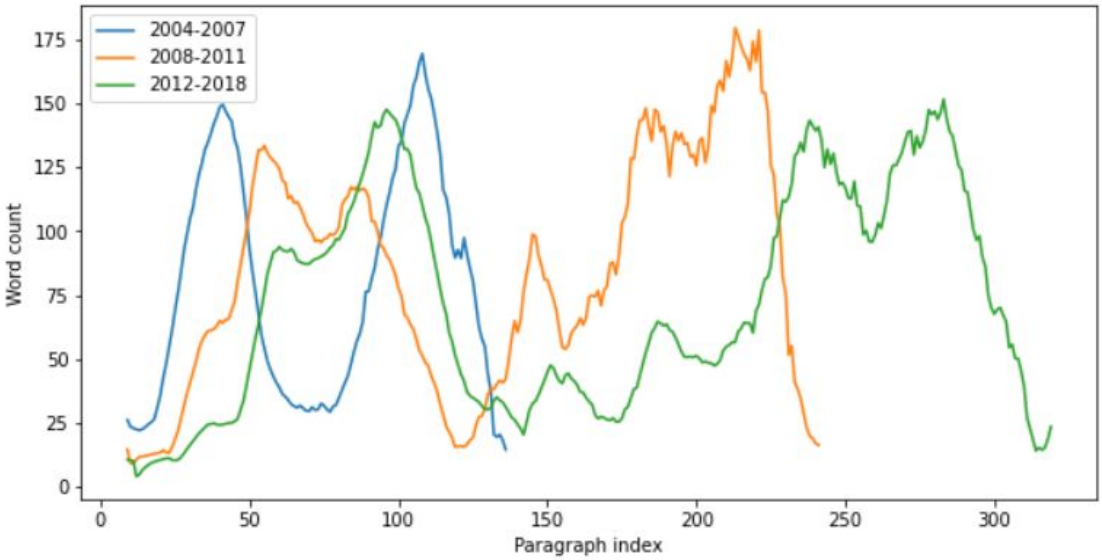
We also observe that the minutes are usually split into 4 following sections: (see Background on FOMC Meeting Minutes)

1. Introduction
2. Economic and financial information
3. Participants' views on developments
4. Policy decisions

The first set of introductory paragraphs contains a list of attendees and procedural items, which is not useful to this project and hence needs to be removed.

Next, we tackle the issue of knowing where the minutes start from.

Figure 2



To see how the minutes are structured, we plot a moving average of words per paragraph (figure 2). The peaks and troughs of this moving average are indicative of new sections, where usually the start of the section includes a header or a very few words.

For simplicity, we have identified three different types of structures that span from 2004-2007, 2008-2011 and 2012-2018 respectively. It is quite clear that the early minutes (2004-2007) began the second section at paragraph 30-40, whilst those minutes later on began at paragraph 50 or onwards.

3.3 Crop Sections

Given the location of where sections roughly start, the easiest way to split the minutes is to find the most common words. Based on our analysis, the minutes will start with phrases similar to 'Staff Review of the Economic Situation' or 'The information reviewed' and end with 'At the conclusion of this meeting' or 'The Committee voted to authorize'.

After we filter out relevant information, total number of paragraphs, sentences and words are reduced by 30-60%, summarized as below:
Total number of files: 131
Number of paragraphs: 5422
Number of sentences: 24232
Number of words: 733281

3.4 Treasury Rates

US Treasury rates represent the market yields of the debts (bill and bonds) issued by the US Treasury Department. These yields are essentially the "price" of these debts. Yields are usually different for debts maturing at different times. Market practitioners mostly focus on 3-month, 2-year, 5-year and 10-year maturities. With yields from different maturities, we can build a yield curve.

Yield slope represents the difference between short term (usually 3-month) and long term (usually 10-year) yields (10y yield minus 3m yield). The change of slope indicates market sentiment of Fed monetary policy, which is based on forecast of future economic conditions. When change of slope is positive, it is called yield curve steepening. Otherwise, it is called flattening.

We use 5-day (a trading week) changes of difference between short-term (3-month) and long-term (10-year) yields (10y yield minus 3m yield) as target variable. The shorter duration might be affected by supply and demand (market fraction) and longer duration might be affected by fundamental economic factors. 5-day is a reasonable duration to capture appropriate market information.

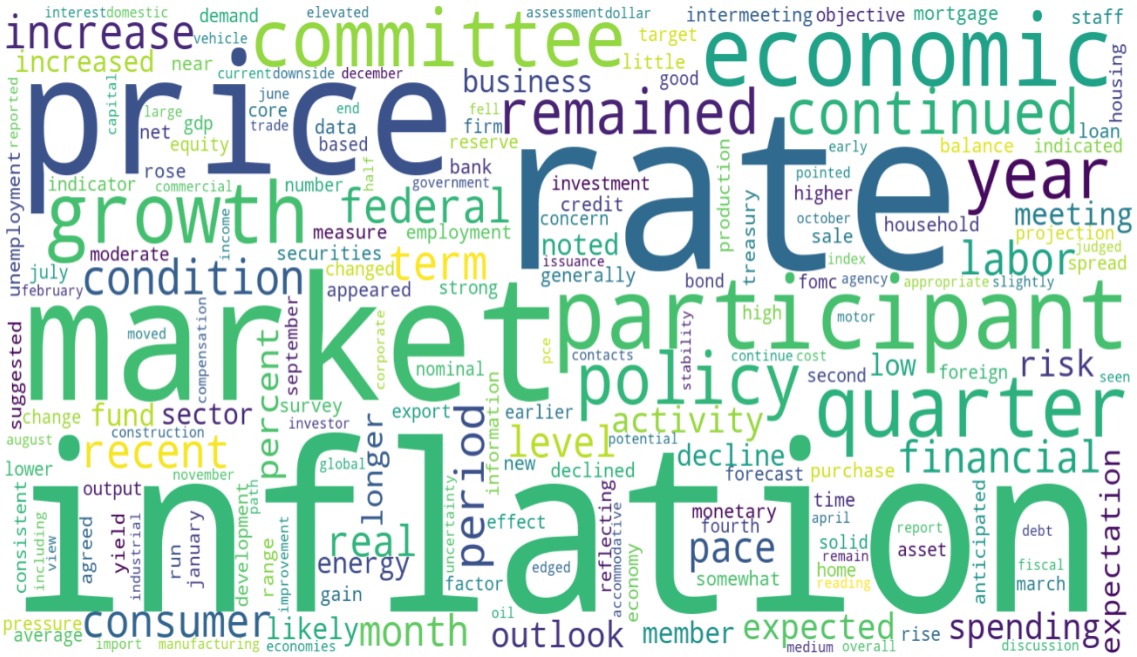
4. Data Preprocessing -- Tokenization

To enhance topic modelling output, we remove stop words and reduce inflectional forms of words back to its roots - using techniques such as

stemming and lemmatization. This process is implemented with customized simple tokenizer.

The Wordcloud chart below (Figure 3) summarizes the top words from the meeting minutes. Not surprisingly, it visually displays the two mandates Fed has: controlling inflation and maintaining growth and employment.

Figure 3



5. Topic Modelling

Now that we have clean text corpus, we can create a bag of words and pass it into a topic model, like Non-negative matrix factorization and Latent Dirichlet Allocation model (LDA). These probabilistic models are ideal for the analysis as each paragraph is distributed amongst a set of topics, where topics themselves are defined by words that frequently appear with each other.

For example, the phrase "Labor market conditions continued to strengthen in recent months, with the unemployment rate declining further and payroll gains well above a pace consistent with maintaining a stable unemployment rate over time" is one that belongs to the labor market category. Additionally, one might say words like labor, payroll and

unemployment are more likely to appear in the same document (or in this case paragraph).

5.1 Non-negative matrix factorization (NMF)

We first conducted NMF analysis. It is an algorithms in multivariate analysis and linear algebra where a matrix V is factorized into two matrices W and H, with the property that all three matrices have no negative elements. We utilize this method to derive word count.

The result shows the average cosine similarity of all paragraphs is 0.4807.

5.2 Latent Dirichlet Allocation (LDA)

LDA is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.

One of the most important inputs in the LDA model is the number of topics (n_components). Given that this is an unsupervised task, there is no best way to choose this input other than trial and error - which in this case results to n_components = 6.

The result shows average cosine similarity of all paragraphs is 0.2977 which is smaller than NMF. As a result, we choose LDA as the topic modeling method.

From Figure 4, we observe that the cosine similarity in each minutes were decreasing after 2008, but took up trend again after 2016. The period of 2008-2016 happens to be the period Fed kept low interest rates.

Figure 4

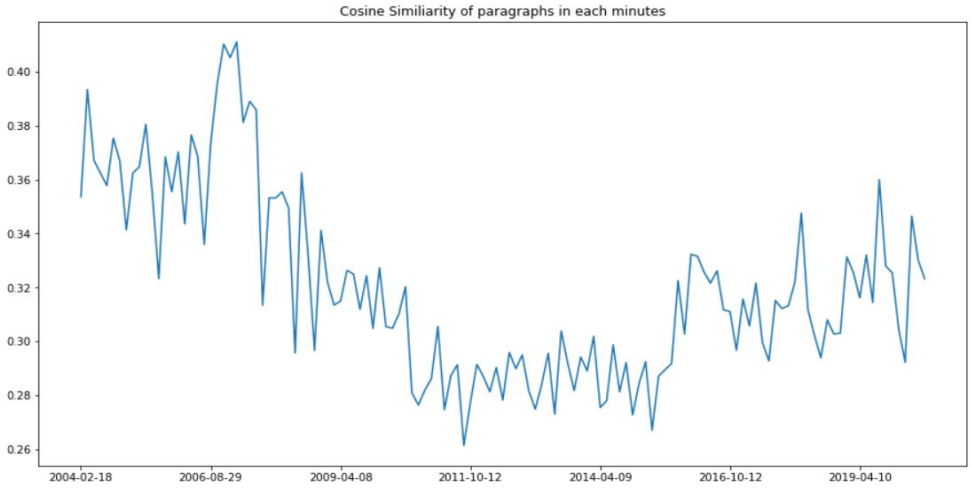


Figure 5 illustrates the top 10 words associated to each topic. Some topics are more distinguishable than others, where words such as inflation, price and energy are usually associated to the topic 'Inflation', whilst others are less so.

Figure 5

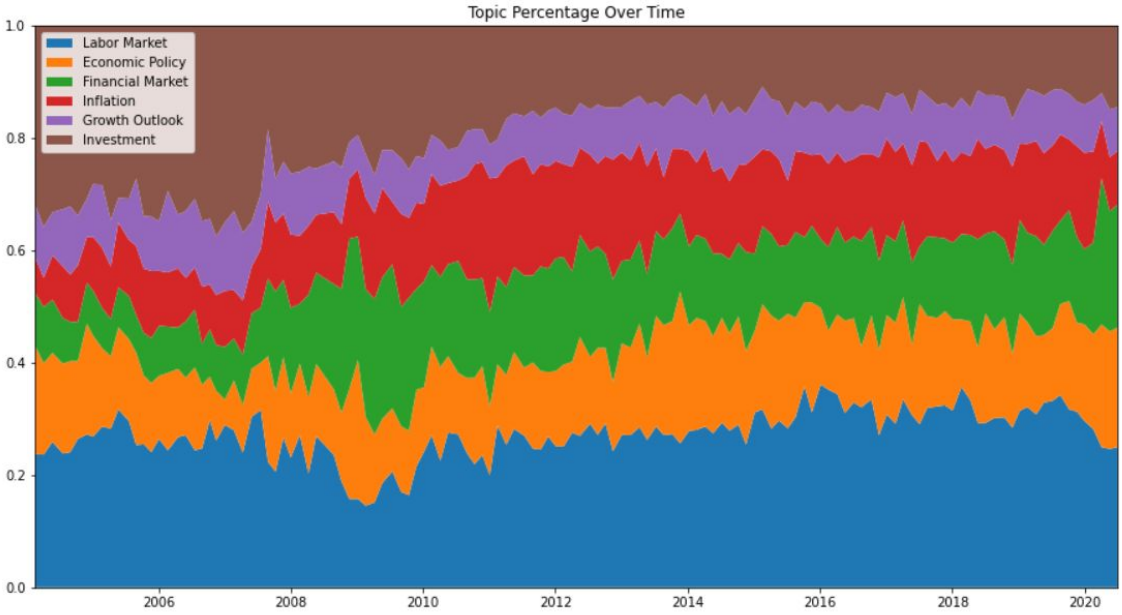
	Topic 0:	Topic 1:	Topic 2:	Topic 3:	Topic 4:	Topic 5:
0	prices	committee	real	period	inflation	quarter
1	remained	federal	quarter	yields	participants	business
2	inflation	policy	consumer	intermeeting	economic	spending
3	consumer	rate	gdp	financial	growth	sector
4	loans	funds	spending	market	labor	production
5	credit	market	growth	treasury	outlook	rate
6	continued	members	second	foreign	market	sales
7	price	conditions	income	markets	recent	continued
8	months	target	forecast	spreads	expected	remained
9	survey	range	pace	equity	prices	months

We also analyzed the shift of topic weights (Figure 6). It looks reasonable. After Global Financial Crisis, the stability of 'Financial Market' gained more attention from the Fed.

To date, the largest topics are 'Economic Policy' and 'Financial Market'. Considering the sharp turn-around on monetary policy and volatile financial market in the first quarter of 2020, this outcome is within expectation.

We also notice the smallest topic currently is 'Economic Growth'. It doesn't necessarily mean Fed cares less about this topic. This is probably due to certain overlap between this topic and other main topics. Again, the topics are not mutual exclusion between each other.

Figure 6



6. Sentiment Analysis

The sentiment in financial market usually shows if investors are optimistic. In this particular context, the sentiment represents if Fed is hawkish (raise rates) or dovish (lower rates).

Two sets of dictionaries were used in this section:

1. Havard IV-4 Psychosociological
2. Loughran and McDonald

The former is less tailored to financial statements, whilst Loughran and McDonald is adapted to include words from 10-K documents (annual financial reports of publicly listed companies).

Given the lack of labelled data, a simply approach is used to calculate sentiment tone:

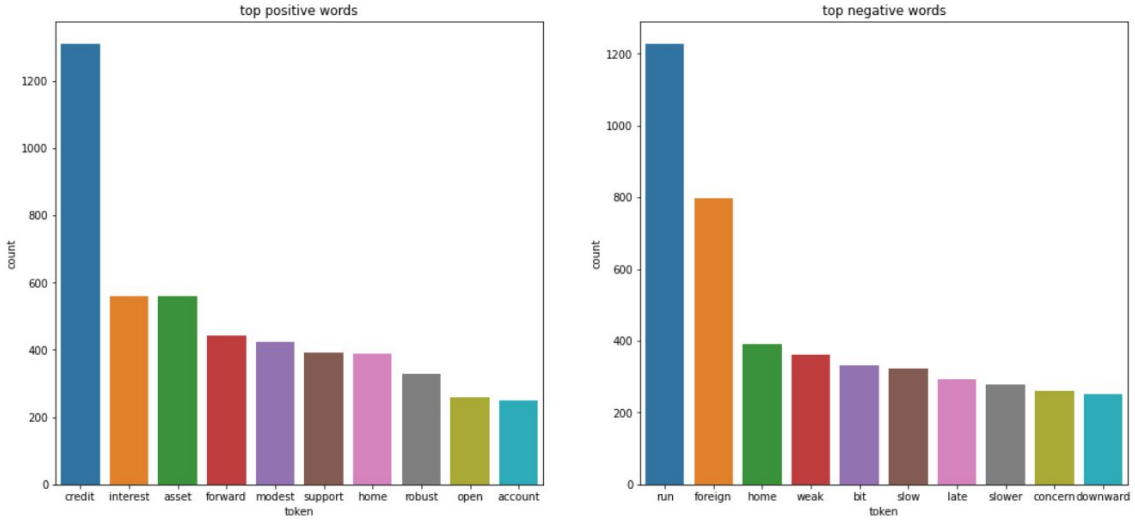
$$Net\ Tone\ = \frac{\#Positive\ words - \#Negative\ words}{\#Positive\ words + \#Negative\ words} \times \frac{1}{\#Total\ words}$$

Where:
Net tone > 0 points to a positive tone
Net tone < 0 points to a negative tone
Net tone = 0 points to neutral tone

6.1 Dictionary Approach

Figure 7 illustrates the top 10 positive and negative words that appear in the corpus. Certain words clearly express strong sentiment, such as "support", "robust", "weak" or "slow". Others might not sound so intuitively. The reason is that official wording, especially that from regulators, tends to avoid directly influencing the financial markets. People need to read between the lines. This "feeling" may require years of markets experience for a human being to gain. However, NLP may be able to learn it in a fast and systematic way.

Figure 7



Number of positive words in corpus: 11269
Number of negative words in corpus: 9218

6.2 Sentiment Extraction

We calculate the total sentiment and sentiments for each topic for each paragraph by multiplying total sentiment scores with topic weights. Then we aggregate scores of each paragraphs to get scores for the entire minutes.

Figure 8 illustrates the changes of sentiments over years. The scores were different by topics showing FED might have different sentiments and prospects for each topic.

The downside of using such a simplistic approach in calculating sentiment is that it can produce some anomalies, in particular for the 'inflation' topic. The word 'inflation' is negative, but context is very important when classifying sentiment - in this case the sentence 'Inflation slowed down' should be a positive phrase but is classified as negative. Nevertheless, we can observe a joint dip amongst all topics in 2008, relating to the financial crisis.

Figure 8

7. Modeling

In this section, we build baseline models and develop more sophisticated models.

Since we have limited amount of data (131 minutes), we use a single paragraph of a minutes as input, there are over 5000 paragraphs in total. The target is if the slope of the yield curve steeper or flatter in 5 days (a trading week). The drawback of using paragraphs are: First, there are duplicated target labels because there is only one label of each minutes and there will be the same for all paragraphs of the minutes. Second, a single paragraph may not contain enough information to predict the target. Both drawbacks will be addressed in minutes level modeling.
We split the data into train and test sets. We will compare the accuracy from both baseline and more sophisticated models.

7.1 Correlations

We investigate the correlation between minutes level topic sentiments and the change of slope. We use spearman rank correlation to avoid the outliers. We only found investment significant at 10% level. Thus, the sentiment scores might not provide much information to predict our target and we did not use them in our models.

```
print(scipy.stats.spearmanr(combined_df['Total'], combined_df['slope_change']))
print(scipy.stats.spearmanr(combined_df['Labor Market'], combined_df['slope_change']))
print(scipy.stats.spearmanr(combined_df['Economic Policy'], combined_df['slope_change']))
print(scipy.stats.spearmanr(combined_df['Financial Market'], combined_df['slope_change']))
print(scipy.stats.spearmanr(combined_df['Inflation'], combined_df['slope_change']))
print(scipy.stats.spearmanr(combined_df['Growth Outlook'], combined_df['slope_change']))
print(scipy.stats.spearmanr(combined_df['Investment'], combined_df['slope_change']))
```

```
SpearmanrResult(correlation=-0.04363422251653285, pvalue=0.6261854316188378)
SpearmanrResult(correlation=0.01968593321994634, pvalue=0.8261227356227164)
SpearmanrResult(correlation=-0.0577864164727175, pvalue=0.5187194782238456)
SpearmanrResult(correlation=0.13536715597626497, pvalue=0.12916053387218715)
SpearmanrResult(correlation=-0.11904130689371126, pvalue=0.18252904205609696)
SpearmanrResult(correlation=0.014275231038809304, pvalue=0.8734394736345492)
SpearmanrResult(correlation=-0.17154005826791635, pvalue=0.053809921638164886)
```

7.2 Baseline Models

The baseline models include word count, Tfidf, NMF and LDA. They represent fundamental natural language process methodologies, which analyze word counts and topics of documents.

Tables below summarize the baseline model's accuracy and classification reports. The test accuracy is from 51-63%. Notably, word count methods

generate better accuracy, while topic models are not promising. The accuracy level of around 63% is the benchmark, which sophisticated models target to beat.

Result of Word Count Model

```
Count Train accuracy:  0.8572207084468665
Count Test accuracy:  0.6255562619198983
              precision    recall  f1-score   support

       False       0.69      0.71      0.70       978
        True       0.51      0.48      0.49       595

    accuracy                           0.63      1573
   macro avg       0.60      0.60      0.60      1573
weighted avg       0.62      0.63      0.62      1573
```

Result of Tfidf

```
Tfidf Train accuracy:  0.8967302452316076
Tfidf Test accuracy:  0.631277813095995
              precision    recall  f1-score   support

       False       0.70      0.71      0.70       978
        True       0.51      0.50      0.51       595

    accuracy                           0.63      1573
   macro avg       0.61      0.61      0.61      1573
weighted avg       0.63      0.63      0.63      1573
```

Result of NMF

```
NMF Train accuracy:  0.5272479564032697
NMF Test accuracy:  0.5104895104895105
              precision    recall  f1-score   support

       False       0.64      0.47      0.55       978
        True       0.40      0.57      0.47       595

    accuracy                           0.51      1573
   macro avg       0.52      0.52      0.51      1573
weighted avg       0.55      0.51      0.52      1573
```

Result of LDA

```
LDA Train accuracy:  0.5583106267029972
LDA Test accuracy:  0.5486331849968213
              precision    recall  f1-score   support

       False       0.66      0.56      0.61       978
        True       0.42      0.53      0.47       595

    accuracy                           0.55      1573
   macro avg       0.54      0.54      0.54      1573
weighted avg       0.57      0.55      0.56      1573
```

7.3 GloVe Embeddings

Beyond baseline models, we first explore word embedding methods, which uses pretrained vectors to represent text inputs.

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

The GloVe model gains accuracy around 56%. It does not outperform baseline models. The results are summarized as below:

```
GloVe Model Train accuracy: 0.5741144414168937
GloVe Model Test accuracy: 0.5562619198982836
              precision    recall  f1-score   support

       False       0.67      0.56      0.61       978
        True       0.43      0.56      0.49       595

    accuracy                           0.56      1573
   macro avg       0.55      0.56      0.55      1573
weighted avg       0.58      0.56      0.56      1573
```

7.4 ELMo Embeddings

ELMo is a deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). It turns out ELMo model underperforms other models, though train accuracy is high. This result indicates that Elmo model overfits the train data set.

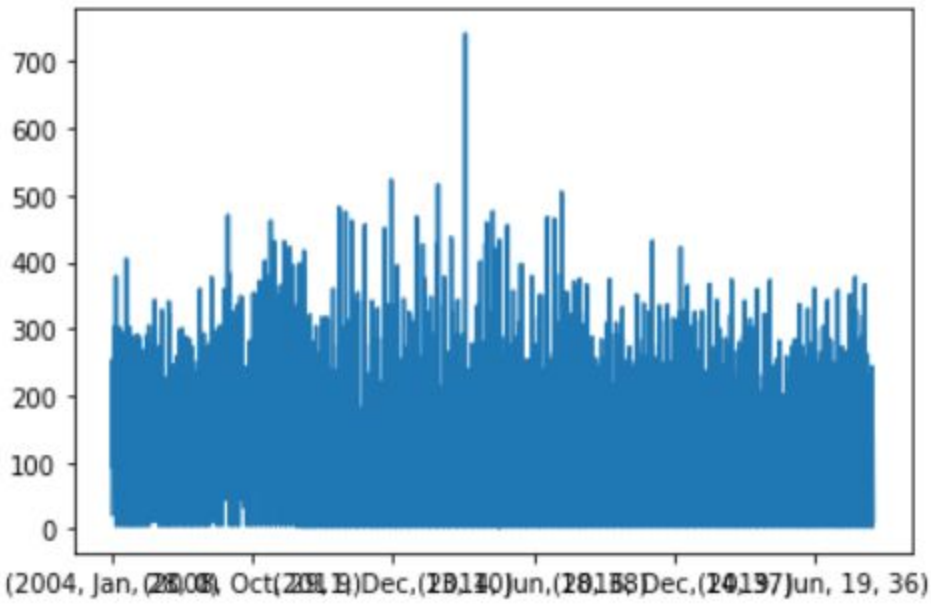
```
Elmo Model Train accuracy: 0.6316076294277929
Elmo Model Test accuracy: 0.54354736172918
              precision    recall  f1-score   support

       False       0.66      0.56      0.60       978
        True       0.42      0.52      0.46       595

    accuracy                           0.54      1573
   macro avg       0.54      0.54      0.53      1573
weighted avg       0.57      0.54      0.55      1573
```

7.5 BERT Embeddings

BERT (Bidirectional Encoder Representations from Transformers) is a technique pre-training NLP, developed by Google. It utilizes huge data to pretrain.

We first check the length of tokens (Figure 9). There is only one paragraph having tokens over 512 and others are below 512 tokens. Thus, it should be safe to truncate the maximum length of token in the BERT model.

Figure 9



The accuracy of Bert model is about 61%. It does not beat baseline models either.

```
Bert Model Train accuracy: 0.5509536784741145
Bert Model Test accuracy: 0.5340114431023522
              precision    recall  f1-score   support

       False       0.65      0.55      0.59       978
        True       0.41      0.51      0.45       595

    accuracy                           0.53      1573
   macro avg       0.53      0.53      0.52      1573
weighted avg       0.56      0.53      0.54      1573
```

Although GloVe, ELMo and Bert are more advanced models (Elmo and Bert incorporate context as well), they still underperform our best baseline model--TFIDF. The reason is probably due to the limited amount of data, which may not fully utilize the power of these models. And since these advanced models are trained on a general dataset and FOMC meeting minutes are a very specialized domain which is a bit difficult to be captured by these pre-train models.

7.6 Minutes Level Embeddings instead of Paragraph Embeddings

The analysis above is based on paragraph embeddings. We further explore the embeddings of whole meeting minutes.

7.6.1 Tfidf

We first utilize Tfidf to generate document level embeddings, as table below.

	raw_text	slope_change	steepen
2004-02-18	The Committee then turned to a discussion of t...	-0.33	False
2004-04-06	The information reviewed at this meeting sugge...	-0.30	False
2004-05-25	The information reviewed at this meeting sugge...	0.03	True
2004-07-21	The information reviewed at this meeting sugge...	-0.29	False
2004-08-31	The information reviewed at this meeting sugge...	0.41	True
...	...	...	...
2019-10-09	Staff Review of the Economic Situation The in...	0.20	True
2019-11-20	Staff Review of the Economic Situation The inf...	-0.01	False
2020-02-19	Staff Review of the Economic Situation The inf...	-0.18	False
2020-05-20	Staff Review of the Economic Situation The cor...	-0.01	False
2020-07-01	Staff Review of the Economic Situation The cor...	-0.06	False

Model based on the Tfidf document level embedding is summarized as below. The result is even worse than paragraph level embeddings.

```
              precision    recall  f1-score   support

       False       0.72      0.54      0.62        24
        True       0.48      0.67      0.56        15

    accuracy                           0.59        39
   macro avg       0.60      0.60      0.59        39
weighted avg       0.63      0.59      0.59        39
```

7.6.2 GloVe and LSTM

Since a minutes consists of a sequence of paragraphs, we also explore LSTM model based on GloVe embeddings with Keras. We used the average GloVe word embedding as the paragraph embeddings and then feed each paragraph embeddings of a minutes into LSTM layer. The outcome is promising and improved to 67%.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 56, 300)           1572900
_________________________________________________________________
lstm (LSTM)                  (None, 300)               721200
_________________________________________________________________
dense (Dense)                (None, 1)                 301
=================================================================
Total params: 2,294,401
Trainable params: 721,501
Non-trainable params: 1,572,900
_________________________________________________________________
None
3/3 [==============================] - 0s 15ms/step - loss: 6.1414 - acc: 0.7500
Accuracy: 0.750
```

	precision	recall	f1-score	support
False	0.69	0.83	0.75	24
True	0.60	0.40	0.48	15
accuracy			0.67	39
macro avg	0.64	0.62	0.62	39
weighted avg	0.66	0.67	0.65	39

In conclusion, we believe that, given the whole minutes as input, an advanced model which could capture the context information of each paragraph could produce better predictions.

8. Deployment

There are three components for our deployment:

Topic model: given a paragraph: it produces the predictions of our six topic percentages in that paragraph,
e.g.
Topic": "{'Inflation': '60.7%', 'Economic Policy': '8.63%', 'Growth Outlook': '29.79%', 'Financial Market': '0.29%', 'Labor Market': '0.29%', 'Investment': '0.29%'}. It could help user to get better understanding the topic and theme of a paragraph.

Sentiment model: given a paragraph, it produces the sentiment of a paragraph (positive or negative) and the scale of the sentiment is between -100% and 100%.
e.g. "Sentiment": "Positive: 17.8571%",

Yield Curve slope movement model, given a paragraph: it produces prediction the impact of the content to the shape of yield curve in 5 days (steepen or flatten) and the probability of that movement.
e.g. Slope": "{'Flatten': '62.61%', 'Steepen': '37.39%'}"

The deployment is based on FASTApi.

9. Limitations and Future Work

As beginners of NLP, we are inspired by this project with a few takeaways.

1. Since there's limited time, we didn't deploy our best model GloVe-LSTM. We need to develop a comprehensive UI and it would be able parse the minutes into paragraphs and show analysis and predictions for both paragraph and entire minutes. Lot of UI works need to be done rather than just relying on the fastAPI website(http://127.0.0.1:8000/docs).
2. Current model is limited to predict treasury rates or yield curves movement. We could try to predict other financial products. Like dollar index, next fed fund rates decisions, etc.
3. Current model is limited to using FED minutes. We could perform similar analysis for other documents of FED. An interesting example would be the speeches of FED. And the practical application would be generating real time predictions (topic, sentiment score, impact of financial products) in real time when a speech was happening. We could also use Bayesian techniques to dynamically update speech level predictions after each paragraph.