Can FOMC Minutes Predict the Federal Funds Rate?

Stanford CS224N Custom Project
March 15, 2020

Ye Ye
Department of Management Science and Engineering
Stanford University
yeye@stanford.edu

Abstract

The Federal Open Market Committee (FOMC) within the Federal Reserve System is responsible for making decisions about the target federal funds rate, and it maintains detailed meeting records, which is called FOMC minutes. In this project, I investigate whether textual information in FOMC minutes can predict directional changes in the federal fund rate in an out-of-sample setting. To this end, I construct hierarchical models that build document embeddings from sentence and word vectors. Document embeddings are then mapped to directional predictions. My results show that models with textual information outperform baseline models using only macroeconomic variables in terms of prediction accuracy for 3- and 6-month forecast horizons.

1 CS 224N logistics

- Mentor: Alexandre Matton
- Collaborators: None
- Sharing project: N/A
- Grading option: 3

2 Introduction

The Federal Open Market Committee (FOMC) within the Federal Reserve System is responsible for overseeing open market operations. It meets eight times a year to make decisions about the target federal funds rate and money supply. FOMC meeting maintains record of minutes, which contain detailed information about policy makers' views on economic outlook and rationales for setting policies. In this project, I seek to answer the following question: Does FOMC minutes have information for predicting changes in the federal fund rates in out-of-sample settings? To this end, I use natural language modeling techniques to extract information from FOMC minutes released at date t and use it to predict the directional movement of interest rates over the period from t to $t + h$.

A major challenge of this project is that while each FOMC meeting record is lengthy, there are few time-series observations as there are only 249 documents released from August 1987 to December 2018. As a consequence, while models with fewer parameters are less susceptible to overfitting this tiny dataset, these models might not be expressive enough to capture the complex dynamics of interest rate movements.

My main methodology can be summarized as follows. Words in sentences are represented using a pre-trained Bidirectional Encoder Representations from Transformers (BERT) [1] model or GloVe [2] vectors. Sentence representations are constructed using one of the three ways: (a) the smooth inverse frequency (SIF) method [3], (b) the BERT vector corresponding to the `[CLS]` token in each sentence, and (c) an one-layer *Word2Sentence* Convolutional Neural Network (CNN) inspired by [4]. Document vectors are obtained from passing sentence vectors through an one-layer *Sentence2Document* CNN. Finally, document vectors are mapped to directional predictions for interest rate movements via a linear layer.

To benchmark performances of NLP models with various configurations, I construct baseline models that use only macroeconomic variables without referring to textual information. My result shows that NLP models outperforms baseline macroeconomic models in terms of out-of-sample prediction accuracy for 3- and 6-month forecast horizons.

3 Related work

There are many recent papers on constructing sentence representations from word vectors. For example, SIF [3] uses a weighted average of word vectors in a sentence to compute the sentence vector. ConvSent [5] constructs a CNN encoder and a Long-Short-Term-Memory (LSTM) decoder to map between a sequence of word vectors and a sentence vector. Skip-thought [6] uses an encoder-decoder architecture to extract a sentence's meaning by reconstructing surrounding sentences. Quick-thought [7] improves on computational efficiency of skip-though by replacing the reconstruction task by a classification task.

There has also been interest in using textual information for making financial predictions. For example, [8] uses a CNN encoder to extract document-level embeddings, and a end-to-end-network is trained to predict movements in the Fama-French factors. In [9], the author applies an attention-based bi-directional LSTM to encode publicly traded companies' 8-K reports, and another attention layer is used to compute event relevancy for predicting stock price movements. In [10], bag-of-words features are extracted from FOMC minutes to predict volatility of interest rates.

On the finance side, there are numerous works on analyzing FOMC records. In [11], the authors use latent semantic analysis (LSA) to extract common themes in FOMC minutes. They find that the themes have in-sample predictive power over Treasury yields. [12] proposes an method to estimate central bank preferences and the implicit inflation target by performing sentiment analysis on FOMC transcripts using a lexical approach. [13] uses Latent Dirichlet Allocation (LDA) to extract topics from FOMC minutes and use this information to estimate the effect of an increase in transparency on communication patterns.

However, to the best of my knowledge, there has not been works on predicting the federal funds rate using FOMC records in out-of-sample settings, which is the focus of this paper.

4 Approach

4.1 NLP models

My overall model is hierarchical and can be divided into the following four modules:[1]

1. Word representation: Represent each word using pre-trained word vectors.
2. Sentence representation: Summarize word vectors in each sentence using a fixed-length sentence vector.
3. Document representation: An one-layer *Sentence2Document* CNN is used to summarize sentence vectors in a document with a document vector.
4. Mapping document vectors to directional predictions: A linear layer maps document vectors to predictions.

The workflow is illustrated in Figure 1. Details of each step are described below.

4.1.1 Word representations

I obtain pre-trained 768-dimensional BERT word embeddings from [14]. GloVe vectors pretrained on *Wikipedia 2014* and *Gigaword 5* are downloaded from the website of [2]. These GloVe vectors have dimensions being 50, 100, 200, and 300. Word vectors are considered fixed during the entire process.

[1] Except for BERT, GloVe, and SIF, I coded everything else in this project by myself.

4.1.2 Sentence representations

For the task at hand, low-dimensional sentence representations can be preferable as the number of time-dimensional observations is too small to support fitting complex models that generalize well. Even though models such as quick-thought and skip-thought achieve state-of-the-art performances on a variety of benchmark tasks, they are not suitable for the specific task at hand as they generate sentence vectors with thousands of entries. Therefore, I consider the following three ways to obtain concise sentence representations:

- **SIF**: Proposed by [3], SIF first calculates a weighted average v_s of embeddings of words in sentence s. Then, matrix X is formed with columns being v_s for sentences s in document $\mathcal{S}$. Sentence s is represented by $\tilde{v}_s = v_s - uu^\top v_s$, where u is the first singular vector of X. According to [3], SIF "achieves significantly better performance than the unweighted average on a variety of textual similarity tasks, and on most of these tasks even beats some sophisticated supervised methods ... including some RNN and LSTM models." The dimension of SIF sentence embedding is exactly the same as that of the constituent word vectors.
- **BERT sentence embedding**: Following [1], before a sentence is fed to BERT, it is padded with the `[CLS]` and the `[SEP]` tokens at the start and the end. Since BERT is bi-directional, the output 768-dimensional `[CLS]` representation can be thought of as a sentence embedding and fed into models for downstream tasks.
- **The *Word2Sentence* CNN**: Inspired by [4], I construct a *Word2Sentence* CNN to extract fixed-length sentence representations from word embeddings. The CNN has one convolutional layer with kernel size being k_s and c_s output channels, which is followed by a max-pooling layer and passing though the ReLU activation function. Unlike [4], which trains word vectors together with the CNN, I use pre-trained word embeddings that are fixed during training.

4.2 Document representation with the *Sentence2Doc* CNN

Recurrent Neural Networks (RNN), including LSTM, can have difficulty dealing with very long input sequences as these models tend to give dis-proportionally higher weights to input vectors near the end of a sequence; however, a lengthy FOMC record can have as many as 471 sentences. I therefore use a *Sentence2Doc* CNN that summarizes sentence embeddings in a document as a "document vector." The CNN has one convolutional layer with kernel size being k_d and c_d output channels. The convolutional layer is followed by a max-pooling layer and the ReLU activation function as before.

4.3 Baseline models with macroeconomic variables

I consider baseline models that use macroeconomic variables instead of textual information to predict interest rate movements. To construct input variables, I use a linear factor model to compress a large cross-section of macroeconomic variables: Given a $T \times N$ panel of macroeconomic data with variables divided into g groups with N_g variables each, I fit a linear factor model with $r = 3$ factors on each group:

$$X_g = F_g \Lambda_g^\top + E_g,$$

where $X_g \in \mathbf{R}^{T \times N_g}$ contains macroeconomic variables in the g-th group, $F_g \in \mathbf{R}^{T \times r}$ and $\Lambda \in \mathbf{R}^{N \times r}$ are the matrices of factors and loading, and $E_g \in \mathbf{R}^{T \times N_g}$ is the matrix of residuals. Loading Λ_g is extracted using only the training set, and factors in the test set are obtained from regressing X_g in the test set on Λ_g. As a result, each cross-section of macroeconomic variable variable at time t is represented by a (rg)-dimensional vector m_t. The following baselines are considered:

1. Logistic regression using m_t
2. Feed-forward neural networks (NN) with 1, 2, and 3 hidden layers that take m_t as input

4.4 Textual models combined with macroeconomic variables

I experiment with concatenating the document vector and macroeconomic vector m_t to form an input variable which is fed to an NN to produce forecasts. This scheme is shown in Figure 2.

5 Experiments

5.1 Data

5.1.1 FOMC Data

FOMC minutes are publicly available online[2]. The authors in [10] have compiled a FOMC corpus covering June 1967 to March 2008[3]. I augment their corpus by including recent data dated until December 2018. Since there was a major monetary policy change in 1987, I start my sampling period in August 1987, following [13]. I pre-process the FOMC dataset by first splitting each document into a list of sentences using the `NLTK` package. I then remove punctuation marks and convert each sentence into a list of lower-case words. Sentences with less than 5 words are removed, and sentences with more than 128 words are truncated. Here are some summary statistics for the pre-processed FOMC dataset:

- Number of documents (August 1987 to December 2018): 249
- Number of words: 1,242,589
- Number of unique words: 16,714
- Maximum / median / minimum length of document: 471 / 168 / 7
- Maximum/ median / minimum length of sentence: 128 / 25 / 5

5.1.2 Macroeconomic variables

I use the monthly McCracken macroeconomic dataset obtained from the website of the Federal Reserve Bank of St. Louis to fit factor models. There are 127 eligible[4] macroeconomic variables divided into 8 categories. Each variable comes with a suggested method (e.g. taking logarithm) for transformation to stationarize the time series. I transform each time series as suggested and then standardize each series by subtracting sample mean and dividing by sample standard deviation. Both sample mean and standard deviation are calculated using only training data.

5.1.3 The effective federal fund rate

The effective federal funds rate is the rate at which depository institutions lend reserve balances to each other overnight without posting collateral. This time series data was downloaded from the website of the Federal Reserve Bank of St. Louis. I calculate 3-, 6-, and 12-month changes in the rate, and the corresponding target variable takes value 1 if there is an increase in the rate and value -1 otherwise. The binary target variable for various forecast horizons are plotted in Figure 3.

5.2 Sample split

I split the sampling period sequentially into training, validation, and test sets. The training and validation periods take 75% and 15% of total observations available each. Since I experiment with forecast horizons longer than a typical time interval between each meeting, observations can have overlaps.[5] To get accurate evaluations of models' out-of-sample performances, I post-pone the starting date of the test set for each forecast horizon such that there is no overlap between the validation and the test sets. I do not remove overlapping observations between the training and the validation set because of the scarcity of data.

5.3 Experimental details

For the *Word2Sentence* and *Sentence2Doc* CNN models, I set kernel sizes as $k_s = k_d = 5$, and number of output channels as $c_s = c_d = 32$. For textual models combined with macroeconomic variable, the final NN that takes both document vector and macroeconomic variable has one hidden

[2]See https://www.federalreserve.gov/monetarypolicy/fomc.htm

[3]https://stanford.edu/ rezab/useful/fomc_minutes.html

[4]I remove variables whose time series start later than 1987.

[5]For example, interest rate changes from Jan. 2020 to Jan. 2021 and from Feb. 2020 to Feb. 2021 are not independent because they have 11 overlapping months.

layer with width 32 followed by ReLU activation function. For baseline NN models with one to three hidden layers, I set the width of hidden layers as 32. Un-normalized logits output by models are converted into probability distributions using the softmax function and cross-entropy loss is evaluated. I apply the Adam [15] optimizer with learning rate 10^{-3} to minimize the cross-entropy loss of predictions. I use early stopping on the validation loss with the patience parameter being set to 10 to decide for how many epochs to train the model. To improve stability, for each configuration, I repeat the training process for 10 times and average the output probabilities on the test set. A model predicts an increase in interest rates if the averaged probability for a rate increase is greater than 50%. I did not do hyperparameter tuning due to time constraint.

5.4 Results

Prediction accuracies on the test set for various model configurations are displayed in Table 1. I note the following observations:

- For 3- and 6-month forecast horizons, textual models out-perform macroeconomic baseline models in terms of prediction accuracy.
- Given textual information, the addition of macroeconomic variable does not improve forecast accuracy.
- Textual models in which sentence representations are extracted by CNN tends to outperform configurations with SIF sentence embedding.
- Performances of the models with BERT embedding and CNN sentence embedding are not stable across forecast horizons. One possible explanation is model instability due to the large number of parameters to be optimized.

6 Conclusion

In this paper, I construct language models to extract information from FOMC minutes and use it to predict directional changes in the federal funds rate. Performances of textual models are compared with those of baseline models which take only macroeconomic variables as input. It is found that textual models outperform baselines in terms of out-of-sample prediction accuracies for 3- and 6-month forecast horizons.

Figure 1: Workflow of textual models

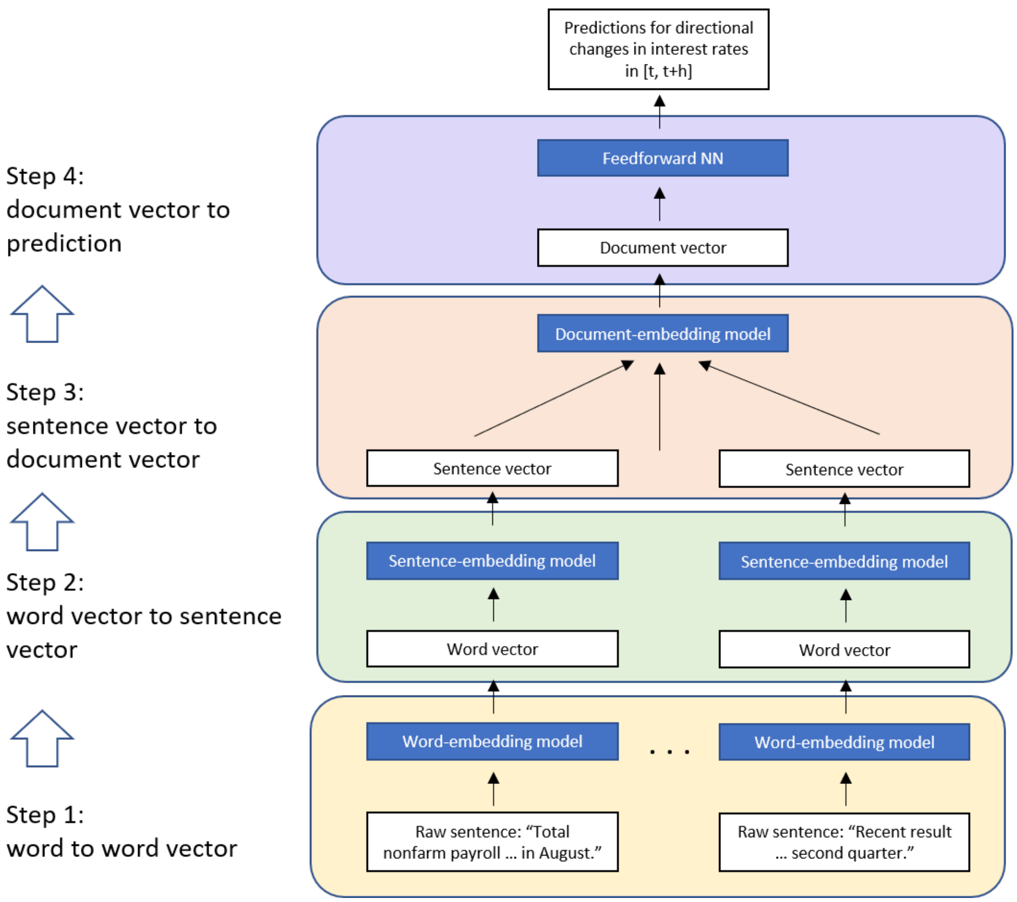


Figure 2: Workflow of textual models combined with macroeconomic variables

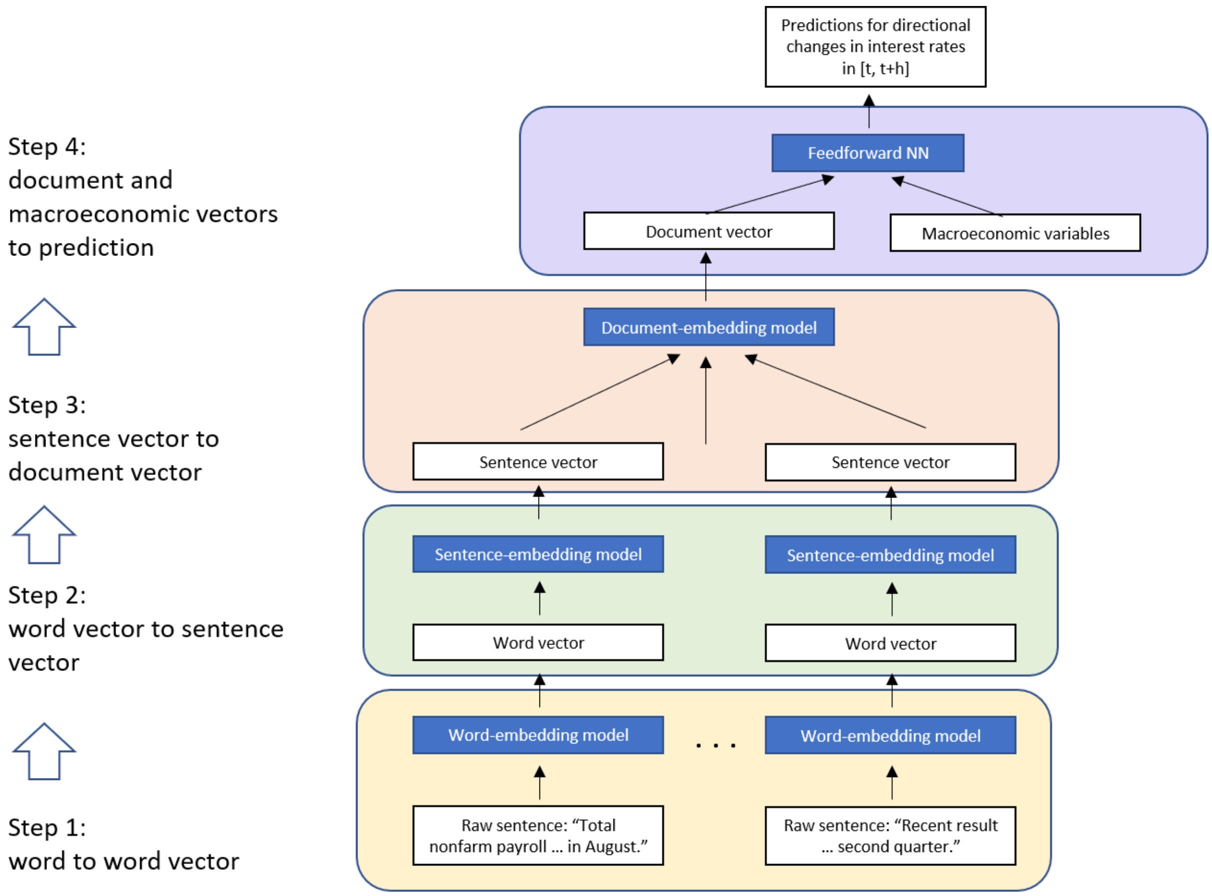
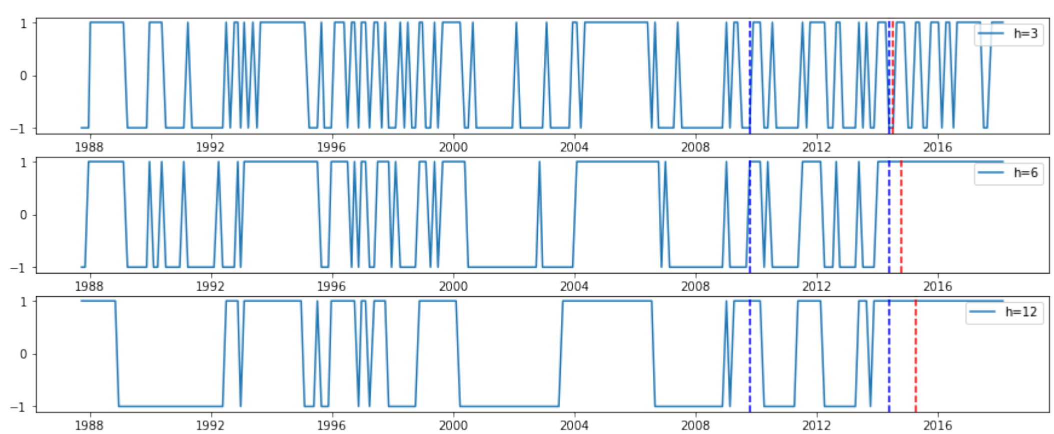


Figure 3: Directional changes in the federal funds rate for 3 to 12-month horizons



Note: The first vertical blue and red lines indicate the starts of the validation and the test periods. The target variable at time t for horizon h takes value 1 if the rate at time $t+h$ is larger than the rate at t. It takes value -1 otherwise.

Table 1: Accuracy rates of out-of-sample predictions for directional changes in the federal funds rate

	3-Month Horizon	6-Month Horizon	12-Month Horizon
GloVe-50D-CNN	0.7	0.8929	**1.**
GloVe-100D-CNN	0.6333	**1.**	**1.**
GloVe-200D-CNN	0.6	**1.**	**1.**
GloVe-300D-CNN	0.6667	0.8929	0.9583
BERT-768D-CNN	**0.7667**	0.3571	**1.**
BERT-768D-Sent	0.7	0.8929	**1.**
GloVe-50D-SIF	0.2667	0.8929	**1.**
GloVe-100D-SIF	0.3	0.	**1.**
GloVe-200D-SIF	0.3	0.1071	0.9583
GloVe-300D-SIF	0.3	0.6786	**1.**
Macro-GloVe-50D-CNN	0.5333	0.9643	0.9583
Macro-GloVe-100D-CNN	0.5333	0.6429	**1.**
Macro-GloVe-200D-CNN	0.5	0.3571	**1.**
Macro-GloVe-300D-CNN	0.4667	0.5714	**1.**
Macro-BERT-768D-CNN	0.5667	0.25	**1.**
Macro-NN-1-Layer	0.4667	0.1786	0.75
Macro-NN-2-Layer	0.4	0.6071	0.7917
Macro-NN-3-Layer	0.4667	0.6071	0.7917
Macro-Logistic	0.4333	0.9643	**1.**

Note: For GloVe-xD-CNN and BERT-768D-CNN, sentence embeddings are extracted using a CNN. For BERT-768D-Sent, the output vector corresponding to the `[CLS]` token in BERT is used as the sentence vector. For GloVe-xD-SIF, SIF is used for computing sentence embedding. For Macro-...-CNN, macroeconomic variables and CNN-extracted sentences embeddings are concatenated and fed into an NN. Models named Macro-NN-x-Layer use only macroeconomic variables as predictors. Macro-Logistic is a logistic regression model fitted only on macroeconomic variables.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[3] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2017.

[4] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *CoRR*, abs/1404.2188, 2014.

[5] Zhe Gan, Yunchen Pu, Henao Ricardo, Chunyuan Li, Xiaodong He, and Lawrence Carin. Learning generic sentence representations using convolutional neural networks. In *EMNLP*, 2017.

[6] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *CoRR*, abs/1506.06726, 2015.

[7] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018.

[8] Andrew Han. Financial News in Predicting Investment Themes.

[9] Mohamed Masoud. Attention-based Stock Price Movement Prediction Using 8-K Filings. pages 1–9, 2014.

[10] Reza Zedah and Andreas Zollmann. Predicting market volatility from Federal Reserve Board Meeting Minutes.

[11] Ellyn Boukus and Joshua V. Rosenberg. The Information Content of FOMC Minutes. *SSRN Electronic Journal*, 2011.

[12] Adam H. Shapiro and Daniel J. Wilson. Taking the Fed at its Word: A New Approach to Estimating Central Bank Objectives Using Text Analysis. *Federal Reserve Bank of San Francisco, Working Paper Series*, pages 01–74, 2019.

[13] Francesco Caselli, Diego Garcia, G Refet, Gilat Levy, Seamus Macgo, Rick Mishkin, Emi Nakamura, Tommaso Nannicini, and Bryan Pardo. Transparency and Deliberation Within the FOMC: A Computational Linguistics Approach. *The Quarterly Journal of Economics*, 133(2):801–870, 2018.

[14] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.