DEPARTMENT OF COMPUTER SCIENCE

Developing a Bespoke Word Embedding for Studying Monetary Policy

Edward Bickerton

MSc Financial Technology with Data Science.

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering.

Tuesday 19th September, 2023

Supervisor: Dr. Xiang Li

Abstract

In setting monetary policy the Federal Open Market Committee (FOMC) produces a plethora of text data across a variety of documents. This project seeks to contribute to the growing body of literature which applies the tools of Natural Language Processing (NLP) to these documents to study the effects of monetary policy. I plan on developing a bespoke word embedding space optimized for downstream NLP tasks involving these FOMC documents. I then hope to apply a state-of-the-art topic modelling algorithm which utilizes the bespoke word embedding. To assess my model I hope to reproduce the results of earlier papers which applied NLP to these documents (described in chapter 2), while using my model in place of their NLP methods. Furthermore, I hope to identify topics whose prevalence in the FOMC documents have significant effects on the responses of the stock market to monetary policy. Finally, to demonstrate the topic models utility I will incorporate it into a trading strategy.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

Edward Bickerton, Tuesday 19th September, 2023

Contents

1 Introduction **1**
1.1 Context . . . 1
1.2 Research Objectives . . . 2

2 Background **3**
2.1 Transparency and Deliberation Within the FOMC . . . 3
2.2 Monetary policy communication, policy slope, and the stock market . . . 3
2.3 Measuring Monetary Policy Shocks . . . 4
2.4 Trillion Dollar Words . . . 5
2.5 Word Embeddings . . . 5

3 Project Execution **9**
3.1 Dataset . . . 9
3.2 Text Preprocessing . . . 10
3.3 FOMC Word Embedding . . . 12

4 Conclusion **13**

A Phrases **16**

B Dataset File Descriptions **20**
B.1 `fomc_documents.csv` . . . 20
B.2 `miscellaneous.csv` . . . 20
B.3 `meeting_minutes.csv` . . . 20
B.4 `bluebooks.csv` . . . 20
B.5 `agendas.csv` . . . 21
B.6 `meeting_transcripts.csv` . . . 21
B.7 `press_conference_transcript.csv` . . . 21
B.8 `greenbooks.csv` . . . 21
B.9 `redbooks.csv` . . . 21
B.10 `policy_statements.csv` . . . 21

List of Figures

2.1 Word2Vec illustration 6
2.2 Feedforward Neural Net Language Model architecture. 7
2.3 Skip-gram Word2Vec architecture. 8

3.1 Visualization of bespoke word embedding using Principal Component Analysis (PCA) to reduce dimensionality. 12

List of Tables

3.1 Data Preprocessing Configurations 10
3.2 Text Quality Statistics 11

A.1 Identified phrases 16

B.1 Text dataset statistics 20

Ethics Statement

This project fits within the scope of ethics application 6683-12073, as reviewed by my supervisor Dr. Xiang Li.

Supporting Technologies

- I used the Scrapy framework [32] to develop a web scraper for the Federal Reserve website.
- I used the Python package `pypdfium2` to extract text data from PDF documents.
- The Python library, Natural Language Toolkit (NLTK) [6] was used extensively for text preprocessing.
- I used the word2vec implementation from the Gensim library [23] along with their phrase detection module.
- I used the Python library scikit-learn [21] for its implementation of PCA.

Notation and Acronyms

CBOW	Continuous Bag-of-Words
CPI	Consumer Price Index
CSV	Comma-Separtated Values
FFR	Federal Funds Rate
FOMC	Federal Open Market Committee
GDP	Gross Domestic Product
HTML	HyperText Markup Language
IRF	Impulse Response Function
LDA	Latent Dirichlet Allocation
LSTM	Long Short-term Memory
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NNLM	Neural Net Language Model
NPMI	Normalized Pointwise Mutual Information
PCA	Principal Component Analysis
PCE	Personal Consumption Expenditures
PDF	Portable Document Format
PLM	Pre-trained Language Model
PMI	Pointwise Mutual Information
POS	Part of Speech
URL	Uniform Resource Locator
VAR	Vector Autoregression

Chapter 1

Introduction

1.1 Context

Monetary policy in the United States is set by a committee within the Federal Reserve (the central bank of the United States): the Federal Open Market Committee (FOMC). Tasked by Congress with the “dual mandate”, the FOMC expand and contract the money supply to maintain both price stability and maximum sustainable employment. Rather than control the money supply directly the FOMC targets the Federal Funds Rate (FFR), this is the interest rate banks (and other depository institutions) charge each other on short-term loans made without collateral. The FFR directly impacts both the interest banks pay to savers along with the interest charged to borrowers. For example mortgage rates tend to rise with the FFR – directly impacting how much money people have to spend. The techniques employed in this project could be used to study the monetary policy of other nations and their respective central banks (provided they release text data). However, I draw my attention to the United States as not only is it the largest economy by GDP, but United States Treasury bonds are frequently used to calculate the risk-free rate of return.

It is hard to deny that the monetary policy set by central banks has a significant effect on the economy. However, the response of economic aggregates such as Gross Domestic Product (GDP), Consumer Price Index (CPI), and Unemployment Rate occur only after a significant lag. Combine this with the complications of endogenous and anticipatory movements of the FFR, and it becomes challenging to study the true effects of monetary policy [24, 3]. The stock market on the other hand appears to be much more responsive; in [5] Bernanke & Kuttner find that an unexpected 25 basis point decrease in the target FFR is associated with a 1% increase in stock indexes.

The FOMC schedules eight meetings per year to discuss their stance on monetary policy and decide on the target FFR. At each meeting a lot of text data is produced. Immediately following the meeting, the FOMC issues a policy statement summarizing their policy decision and economic outlook. Likewise, the Chairman holds a press conference to provide further context to their decision. A set of minutes is published three weeks after the meeting and even complete transcripts of FOMC meetings are released after a five-year lag. In addition, during the press conference following the last scheduled meeting of each quarter, the Chairman discusses the economic projections submitted by each of the FOMC participants. Even still, there is more text data surrounding each meeting in the documents prepared in advance of each meeting. This includes the “Beigebook”, which contains information on current economic conditions, and the “Tealbooks”, which give an economic analysis and description of policy alternatives.

The transparency of the FOMC in publishing this quantity of text based data suggests that the study of monetary policy and its effects stands to benefit from the use of Natural Language Processing (NLP) techniques. Indeed, many papers have already been published applying various NLP methods on FOMC documents to different ends, as described in chapter 2. However, much of this past research relies on dictionary based approaches to NLP which depend on domain level knowledge of economics and monetary policy. This area of research also stands to benefit from the recent advancements to NLP. Therefore, a key benefit of my approach in avoiding the heavy reliance on dictionary NLP approaches is that domain level knowledge of economics or even of the language is not a prerequisite to this type of analysis.

1.2 Research Objectives

In my project I aim to apply state-of-the-art NLP techniques to the text data produced by the FOMC to gain insight into the factors of monetary policy deliberation which drive the responses of the stock market.

Bespoke word embedding. The first stage of my project will be to develop a bespoke language model to generate a word embedding space specialized to the FOMC documents. The recent improvements of word embeddings are due by in large to the seminal papers by Tomas Mikolov et al. introducing the Word2Vec algorithm [17, 16]. This algorithm and others like it described in [31, 1], seek to encode the syntactic and semantic information of a word into a fixed length vector which can then be used in downstream NLP tasks. The power of word embeddings can be demonstrated in the following common example:

$$King - Man + Woman = Queen$$

where the binary operations are performed on the vector representations of each word, as depicted in figure 2.1.

There are high quality open source word embeddings available, for example word vectors developed in [17, 16] were trained on about 100 billion words from a Google News dataset and are available through Gensim [23]. However, I believe there are some key benefits associated with training a dedicated word embedding. There remains domain specific vocabulary not captured by generic word embeddings, such as: acronyms (e.g. FOMC, FFR, and PCE), names which carry a significance (e.g. Powell, Yellen, and, Bernanke), and finally n-grams which aren't frequently used outside of discussions of monetary policy (e.g. "Federal Funds Rate"). Additionally, certain words take a different meaning in the given context, for example *tightening* tends to refer to measures taken by a central bank to slow down economic activity and bring down inflation.

Topic model. Topic modelling is the unsupervised machine learning task of compressing a large corpus of documents into a collection of topics capturing the most prevalent subjects present in the corpus. Each document can then be labelled with the most representative topics. As described in [9] the field of topic modelling has benefited greatly from the incorporation of modern language models such as Word2Vec. For example Latent Feature LDA [20] utilizes word embedding spaces in conjunction with the classical Latent Dirichlet Allocation (LDA) topic model, and CluHTM [30] which utilizes "CluWords", or clusters of words derived from word embeddings to create a hierarchical topic model.

Practical application. To evaluate my topic model I aim to reproduce some results of the papers described in chapter 2. I plan on investigating the sensitivity of assets such as Gold, Oil and Bitcoin to monetary policy, and hope to identify the topics whose prevalence in FOMC meetings has the greatest predictive power on returns. Inspired by the trading strategy of [26] which incorporated their hawkish-dovish classifier, I too am interested in incorporating my topic model into a trading strategy to further validate its utility.

Put plainly, the concrete aims and contributions of this project are:

1. Provide easy access to a comprehensive dataset of text from FOMC documents.
2. Release web scraping code such that the dataset can be updated when new FOMC documents are released.
3. Apply state-of-the-art NLP techniques to said text data — with an emphasis on unsupervised approaches avoiding reliance on domain level knowledge of economics or monetary policy.

Chapter 2

Background

In the following chapter I describe a handful of papers, each of which apply NLP techniques to documents produced by the FOMC with regard to studying the effects of monetary policy decisions. I then conclude by describing the state-of-the-art NLP algorithms I plan on utilising.

2.1 Transparency and Deliberation Within the FOMC

Since the 1970s FOMC meetings have been recorded to prepare the minutes – a somewhat abbreviated record of the discussions, decisions and actions taken during the meeting. Committee members were under the impression that these recordings were deleted, however, transcripts were in fact archived. This was revealed when in 1993 Federal Reserve agreed to publish all past transcripts and release future transcripts with a five-year lag going forward. Hence, transcripts are available for both periods in which policymakers did and did not believe their deliberations would be made public.

Hansen et al. [12] take advantage of this natural experiment to study the effects greater transparency has made on the decision-making process of monetary policy. They focus their attention on two portions of each transcript; the discussion of the economic situation, and the monetary policy strategy discussion referred to as *FOMC1* and *FOMC2* respectively.

To capture the nature of FOMC1 and FOMC2 discussions the researchers apply a probabilistic topic modelling algorithm – Latent Dirichlet Allocation (LDA). In addition to applying standard text preprocessing they apply a Part of Speech (POS) tagger described in [28], then use POS patterns identified in [14] to identify collocations such as “labor market” – creating a single term when a collocation has a frequency above a certain threshold.

They estimate LDA on the set of individual statements in FOMC1 & FOMC2, producing a distribution of topics within individual statements. However, to study the documents composed of the statements, they fix the set of topics and re-estimate the document-topic distribution. An in depth analysis of results can be found in [12], however they find large behavioural responses to the increased transparency.

2.2 Monetary policy communication, policy slope, and the stock market

Federal funds futures are tradeable contracts which enable investors to speculate or hedge against the target FFR decided by the FOMC. Thus, the price of these contracts contains valuable information about the market’s expectations of future monetary policy (*FedWatch* is a handy tool which gives probabilities of changes to the FFR, as implied by the futures price data).

Neuhierl & Weber [19] take advantage of this when they define the slope factor as a difference of differences, specifically it is the change in difference between the FFR implied by the one-month and three-month contracts, from one week to the next. When this is positive it implies expectations of faster future FFR hikes, similarly, a negative slope factor reflects the markets expectations of a faster monetary policy easing. A theoretical justification of this definition of slope factor can be found in [19].

Aside from predicting stock returns, Neuhierl & Weber find that the slope factor correlates with the *tone* of speeches made by FOMC members. To obtain a measure of tone they collect speeches[1] made by

[1] FOMC member speeches are available at: www.federalreserve.gov/newsevents/

the FOMC members and apply a "search-and-count" method as in [2] on each speech. That is, using a dictionary of "hawkish" and "dovish" terms they calculate a *Net Index* using the total number of hawkish and dovish terms in the speech. When they restrict the dataset to speeches made by either the chair or vice chair and include only speeches containing at least one term from the hawk/dove dictionary, they find that speeches explain over 12% of the variation in the slope factor. As expected, hawkish speeches (as implied by the net index) are associated with increases in the slope factor, and dovish speeches with decreases in slope factor.

2.3 Measuring Monetary Policy Shocks

2.3.1 The Price Puzzle

To study the effects of monetary policy on the economy, Sims [27] estimates a Vector Autoregression (VAR) model using data from a handful of countries on a selection of variables including the consumer price index and the short interest rate (for the United States this is the FFR). In this paper innovations (changes to the variable unexplained by the model) in the short interest rate are considered a proxy measure of monetary policy.

When analysing the Impulse Response Function (IRF), a strong positive response of prices to interest rate innovations is found. I.e., according to the VAR model, unexpected monetary tightening leads to a counter-intuitive increase in inflation - contradicting the conventional belief that contractionary monetary policy is deflationary. This phenomenon is commonly referred to as the "Price Puzzle". Sims reconciles this by suggesting that central banks have a better idea of future inflation than what can be obtained from the variables included in the VAR model and because central banks are forward-looking, in anticipation of high inflation they raise interest rates to dampen the expected rise in price level. Hence, prices rise after a contraction due to the anticipated high inflation, though by less than they would have without the contraction.

2.3.2 A New Measure of Monetary Policy

To investigate the true effects of monetary policy Romer & Romer create a new measure of monetary shocks free of both endogenous and anticipatory movements [24]. They claim that the FFR changes from day-to-day for reasons unrelated to monetary policy and that over the Greenspan era (during which the FFR was only weakly targeted by the Federal Reserve), the FFR tended to rise endogenously with economic activity. Thus, the FFR is a biased measure of monetary policy, causing researchers to underestimate the negative effects of contractionary monetary policy on real economic variables.

They argue that using the Federal Reserve's *target* for the FFR as a measure avoids the issue of endogeneity but is still plagued by the anticipatory movements made by the central bank. Thus, Romer & Romer incorporate both; the intended funds rate around FOMC meetings, and the Federal Reserves' internal forecasts found in the "Greenbook", to construct their measure. The full specification of their new measure can be found in [24] but to summarise; they estimate a regression with change in target FFR as dependant variable and various numerical forecasts from the Greenbook as the predictor variables. They then take the residuals from this regression as their measure of monetary shocks.

The idea being that these numerical forecasts summarise the information used by the FOMC to determine the target FFR and that these residuals are purged of systematic actions taken by the central bank in response to information about future developments.

When using a VAR model to investigate the relationship between their new measure and price they find that the response of prices to a 1% innovation in their measure is small, irregular, and insignificant for eight months before slowly decreasing. While the price puzzle is eliminated, the negative effects of a contractionary shock on price level only become significant after around 18 months.

2.3.3 A Natural Language Approach

Nearly two decades later, Aruoba & Drechsel [3] build upon the ideas of Romer & Romer by incorporating more information from the documents prepared for FOMC meetings into the regression used to derive their measure of monetary policy shocks. They do this by constructing sentiment indicators on nearly 300 economic terms via NLP techniques. A regression including only numerical forecasts yields an R^2 of 0.5, the addition of their sentiment indicators (along with non-linearities in the dependant variables) increases this to 0.94, that is to say that the implied exogenous component of the target FFR changes from 50% to 6% (a result more in line with *good* monetary policy).

It's worth noting that despite the goodness of fit, the goal of the regression is not to predict the target FFR decided by the FOMC. Instead, its purpose is to purge the intended funds rate series of movements taken in response to forecasts. In any case the Greenbook is only released five years after the FOMC meetings.

Inspired by [13], their approach to NLP into two main steps. First, they extract singles, doubles (pairs of adjacent words) and triples from the documents and calculate their frequencies. Terms are ordered by frequency and the authors move down the list and select terms which they consider to be economic concepts. Finally, to construct their sentiment classifiers they consider each occurrence of their selected economic concept and look at the 10 words preceding and following the occurrence. To identify the sentiment of these 20 words they use a dictionary of positive and negative terms constructed in [15] specifically for financial text, each positive term adds one to the indicator while a negative term subtracts one.

They also describe a real-time version of their measure which applies the same methods outlined above on a subset of the documents prepared for the FOMC meetings, namely the Beigebooks which are made available before each meeting.

2.4 Trillion Dollar Words

A Pre-trained Language Model (PLM) is a type of deep learning model that has been *pre-trained* on a large amount of text data. Pre-training commonly involves training the model to predict a missing word in a sentence given the surrounding context. Rather than being optimised for a specific NLP task these models learn general language patterns and grammar, they can then be fine-tuned for a specific NLP task.

While pre-training PLMs requires a large upfront investment of computational resources, the fine-tuning stage requires much less labelled data than training a model from scratch for a specific task. Thus, for the novel task of hawkish-dovish classification where labelled data is hard to come by; PLMs make for an ideal candidate. Indeed, in [26] Shah et al. show that a fine-tuned PLM outperforms both Long Short-term Memory (LSTM) neural networks (a variant of a recurrent neural network which are designed for sequential data such as language), and a traditional rule-based method relying on a hawk/dove dictionary [11].

However, the fine-tuning of PLMs still requires *some* labelled data. For this Shah et al. sample roughly 2,500 sentences from FOMC meeting minutes, speeches made by FOMC members, and press conference transcripts; the sentences are manually labelled as either “Hawkish”, “Dovish”, or “Neutral” by two annotators to reduce potential labelling bias. On this dataset the rule-based classifier achieved an F1 score of roughly 0.5, the LSTM performed similarly if not worse (perhaps due to the limited labelled data), and the top performing PLM based model (RoBERTa-large) achieved an F1 score of 0.7. Surprisingly, ChatGPT without any fine-tuning (zero-shot) also outperformed both the LSTM and rule-based classifiers (but not the RoBERTa-large based model).

From their PLM based hawkish-dovish classifier Shah et al. construct a document measure of hawkishness. They find that this measure of monetary policy stance is positively correlated with both inflation and treasury yields. They also find that a simple trading strategy of shorting the QQQ index when a FOMC press conference is hawkish (according to their measure) and opening a long position when an FOMC press conference is dovish, outperforms the simple buy and hold strategy. Further validating the utility of their new measure.

2.5 Word Embeddings

Word similarity. In the field of text analysis words are frequently treated as atomic units, represented plainly as indices in a vocabulary. While this choice of representation has the advantage of simplicity, the notion of similarity between words is absent. If instead words are represented by a continuous vector, $\boldsymbol{w} \in \mathbb{R}^D$, the similarity of two words, represented by vectors: $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$, can evaluated using an appropriate similarity metric defined over the vector space. A common choice for word vectors is cosine similarity. That is taking $\cos(\theta) \in [-1, 1] \subset \mathbb{R}$, where θ is the angle between the two vectors. Thus, a cosine similarity of 1 would imply the words are very similar in meaning (their corresponding vector representations have the same direction), likewise a cosine similarity of -1 corresponds to semantically opposite words. In practice this is computed via:

$$\cos(\theta) = \frac{\boldsymbol{w_1} \cdot \boldsymbol{w_2}}{||\boldsymbol{w_1}||\,||\boldsymbol{w_2}||}, \tag{2.1}$$

where $\cdot$ is dot product and $||\cdot||$ is the magnitude; only the vectors non-zero values need be considered making cosine similarity highly efficient for sparse vectors – essential when vocabularies are large. Likewise, magnitude invariance of the metric is beneficial as direction takes precedence in a space where each dimension encodes a different linguistic regularity.

Word vector arithmetic. In fact not only do high quality word vectors exhibit closeness of semantically similar words, but in [17], Mikolov et al. discover that word vectors can be used to solve complex similarity tasks. For example, they answer the following question:

> "What is the word that is similar to *king* in the same sense as *woman* is to *man*?",

using vector arithmetic (as depicted in figure 2.1) by calculating $X = King - Man + Woman$, then searching the vector space for the word closest to X (with respect to cosine similarity). Which, when the vectors are well constructed, returns the vector corresponding to the word *Queen*. This method extends beyond the simple masculine-feminine word relationship to a diverse set of both semantic and syntactic relationships – consider the questions:

i) "What is the word that is similar to *London* in the same sense as *France* is to *Paris*?" &

ii) "What is the word that is similar to *small* in the same sense as *biggest* is to *big*?",

i) tests the word vectors against the semantic relationship of a capital city to its encompassing country, while ii) tests for the superlative relationship which is syntactic in nature. In [17] Mikolov et al. develop a comprehensive set of nearly 20,000 such questions divided into four semantic and nine syntactic categories, they can be used to evaluate the quality of word vectors across multiple degrees of similarity.

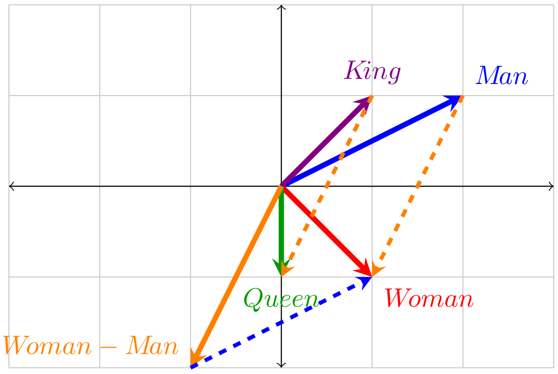

Figure 2.1: Capturing semantic relationships through vector arithmetic.

2.5.1 Algorithms

Feedforward Neural Net Language Model (NNLM)

The idea that hidden units of a neural network, distinct from both the input and output, come to represent important features of the input dates back to 1986, when the backpropagation algorithm was popularised in [25]. This was first put to use in estimating word vectors in 2003 when Bengio et al. [4] proposed the feedforward NNLM architecture depicted in figure 2.2. The neural network learns the conditional probability function over the entire vocabulary given the previous N words ($N = 3$ in figure 2.2) by maximising the log-likelihood of the training data.

The previous N words are first mapped to a vector $\boldsymbol{w}_{word} \in \mathbb{R}^D$ at the projection layer. This is done by first encoding words as one-hot vectors $\in \{0,1\}^V$, that is, a vector of the same length as the vocabulary and whose values are all 0 except for the i-th term which has a value of 1, where i is the index of the word in the vocabulary. The projection layer then consists of multiplying the input vectors by $W \in \mathbb{R}^{D\times V}$ whose values are learnt during training. Note: the i-th column of W is the word vector corresponding to the i-th word in the vocabulary.

The resulting N word vectors are then concatenated and the $N \times D$ values passed through a fully connected non-linear hidden layer of size H and finally an output layer of size V with softmax activation function – ensuring the output is a probability distribution over the vocabulary. That is, the i-th component of the output approximates the probability of word i following the previous N words, for example, in figure 2.2 we might expect the output nodes associated with words such as *strangers*, *cars*, and *cats* to score highly while *rainbow* and *kindness* take values close to 0.

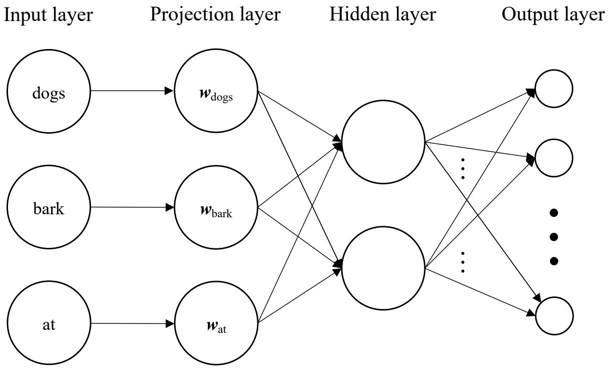

Figure 2.2: Feedforward Neural Net Language Model architecture.

2.5.2 Word2Vec

The main focus of [4] and the feedforward NNLM was not to learn high quality word vectors, this was instead a means to an end of approximating the joint probability function of sequences of words. In [17, 16] Mikolov et al. develop two neural network architectures, Continuous Bag-of-Words (CBOW) and Skip-gram, jointly referred to as Word2Vec. Optimised for learning high-quality word vectors, these models achieve large improvements in accuracy (with respect to the task defined in [17]) while reducing training time. In short, this is accomplished by employing simplified architectures that can be trained on a text corpus that is several orders of magnitudes larger than was previously feasible.

Training complexity. Defining training complexity as the number of parameters that need to be accessed to fully train the model, the feedforward NNLM has a training complexity proportional to

$$\underbrace{N \times D}_{\text{Projection layer}} + \underbrace{N \times D \times H}_{\text{Hidden layer}} + \underbrace{H \times V}_{\text{Output layer}}, \tag{2.2}$$

where N is the number of previous words taken as input, D is the dimension of the learned word vectors, H is the size of the hidden layer, and V is the size of the vocabulary. While $H \times V$ is the dominating term it can be avoided by, for example, approximating the softmax by the hierarchical softmax [18] which utilises a binary tree representation of the vocabulary to reduce the number of output nodes which need to be evaluated to $\log_2 V$. As a result, the majority of complexity comes from the hidden layer – associated with the term: $N \times D \times H$.

Continuous Bag-of-Words

The CBOW Word2Vec architecture [17] can be thought of as a heavily stripped down version of the feedforward NNLM. They both output a predicted target word based on an input of context words, however for CBOW, this consists of a few words before and *after* the target word. First, the N context words are mapped to their associated word vectors $\boldsymbol{w}_{word} \in \mathbb{R}^D$, as in 2.5.1. These N vectors are then averaged element-wise and the resulting D values are passed *directly* to the hierarchical softmax output layer. Resulting in a training complexity proportional to

$$\underbrace{N \times D}_{\text{Projection layer}} + \underbrace{D \times \log_2 V}_{\text{Output layer}}. \tag{2.3}$$

By taking the element-wise average of the input word vectors as opposed to concatenating them as in 2.5.1 word order is disregarded, hence the name; continuous *bag-of-words.*

Skip-gram

Unlike CBOW, the skip-gram architecture predicts the context words surrounding an input word based on its vector representation. Each context word is predicted by passing the input words associated vector representation to a hierarchical softmax output layer as in figure 2.3.

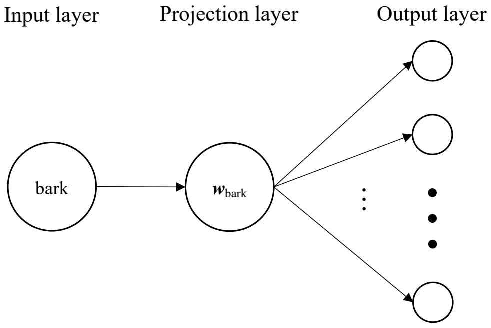

Figure 2.3: Skip-gram Word2Vec architecture.

Skip-gram predicts N preceding and N subsequent words for each training word, where N is uniformly sampled from $[1, C] \subset \mathbb{N}$ and C is a hyperparameter controlling the maximum distance of the context words. This sampling mechanism ensures that words closer to the input word are sampled more frequently than distant (and thus less relevant) words. The training complexity of the skip-gram is proportional to

$$C \times \left(\underbrace{D}_{\text{Projection layer}} + \underbrace{D \times \log_2 V}_{\text{Output layer}} \right). \tag{2.4}$$

The skip-gram, while taking longer to train, produces higher quality word vectors than CBOW with respect to the semantic questions discussed in [17], especially when the training set is smaller.

Chapter 3

Project Execution

3.1 Dataset

The first and prerequisite task in completing this project is constructing a dataset of text produced by the FOMC. The Federal Reserve website[1] is home to a breadth of document types surrounding monetary policy from 1936 to the present. For example, some documents are prepared prior to each meeting to assist the decision-making process while others record the events of the meeting (either in summary or verbatim). Additionally, some documents are released shortly after meetings to announce monetary policy decisions. Unsurprisingly, the nomenclature, content, and release schedule of these documents has evolved over time, further complicating the construction of the dataset.

3.1.1 Data structure

Along with the text itself, for each document I record the following:

- kind (the name given on the Federal Reserve website),
- meeting date (for Beige Books where this is not given, I use the soonest subsequent meeting),
- release date (when not given, this is inferred according to the release schedule described at [10]), &
- Uniform Resource Locator (URL).

I opt to store the data in the highly interoperable Comma-Separtated Values (CSV) format. There are over 30 document kinds in the final dataset due in large part to documents of similar content being rebranded. Such as the *Redbooks*, (officially titled "Current Economic Comment by District") which was reformulated into the *Beige Book* (officially titled "Summary of Commentary on Current Economic Conditions by Federal Reserve District") in June 1983. To simplify this I group together documents I deem to be of same *type* into nine separate CSV files, the details of which can be found in appendix B.

3.1.2 Web Scraping

Since there are thousands of documents across dozens of webpages in both HTML and PDF formats, collecting the text manually is impractical. Hence, I use web scraping to automate the collection process. Specifically I implement four web crawlers using the Scrapy framework [32], each starting on a different webpage. Further details including the web scraping code is available on my GitHub account in the Fed-Scraper repository.[2]

A key benefit of this approach to data collection is that updating the dataset to include documents from the latest FOMC meeting requires only to run some code. The dataset used in the remaining sections is available on Kaggle[3] and consists of 5,958 documents made up of nearly 63 million words from a vocabulary of over 375 thousand unique tokens.

[1] The Federal Reserve website: www.federalreserve.gov

[2] Fed-Scraper GitHub repository: https://github.com/rw19842/Fed-Scraper

[3] FOMC text dataset available at: www.kaggle.com/datasets/edwardbickerton/fomc-text-data

3.2 Text Preprocessing

Raw text data is often noisy of low quality. For example stopwords (words such as "are", "of", and "the") add little meaning and yet from table B.1 we see that they make up over 30% of the FOMC text dataset, and almost 40% of the transcripts. Furthermore, standardising the text, by for example capitalisation normalisation and/or lemmatization, reduces the size of the vocabulary which in-turn improves the performance of downstream NLP tasks. In particular, text preprocessing has been shown to have significant effects on the performance of topic models and word embeddings [8, 22].

To illustrate what text preprocessing entails, consider the following sentence taken from the FOMC press conference held on the 21st of March 2018 along with the corresponding output from my heavyweight configuration defined in table 3.1:

> "The unemployment rate remained low in February at 4.1 percent, while the labor force participation rate moved higher."
>
> ["unemployment_rate", "remain", "low", "february", "percent", "labor_force", "participation", "rate", "move", "high"]

3.2.1 Preprocessing Configurations

I adopt the terminology defined in [8], wherein a preprocessing *rule*, r_l, is an operation that changes or removes a token, and a *configuration*, $C_k = r_1, r_2, \ldots, r_k$ is a sequence of rules which are applied to a Document. I implement the following three configurations defined in table 3.1: baseline, lightweight, and heavyweight, applying more thorough preprocessing with each subsequent configuration as the names suggest.

Table 3.1: Data Preprocessing Configurations

	Baseline	Lightweight	Heavyweight
Sentence	remove urls	——	——
	remove accents	——	——
	capitalisation normalisation	——	——
		expand contractions	——
Word	remove punctuation	——	——
	remove numbers	——	——
		lemmatization	——
		remove stopwords	——
		remove short words	——
			n-gram creation

3.2.2 Implementation

Following the spirit of Churchill and Singh's `textPrep` [8] I implement my preprocessing pipeline in a modular fashion such that it can be extended with additional preprocessing rules and that creating a new configuration is as easy listing the desired rules. My text preprocessing code is available in the FOMC-text-preprocessing GitHub repository.[4]

My methodology differs from that of [8], in that rather than tokenize documents on white-space, thus producing a series of tokens for each document $D_i = \{d_1, d_2, \ldots, d_n\}$, I first tokenize the document based on sentences and then on words. Hence, producing a list of lists, $D_i = \{s_1, s_2, \ldots, s_p\}$ where each sentence is a sequence tokens, $s_i = \{w_1, w_2, \ldots, w_q\}$. This has the benefit of maintaining the sentence structure of a document (which is particularly valuable when training a word embedding) while allowing the flexibility to apply certain preprocessing rules to the larger sentence string and others to each word. Rules which apply to sentences as opposed to words are differentiated in table 3.1, although this is largely down to my specific implementation and not intrinsic to the rules themselves. I use NLTK's `tokenizer` package to perform both the sentence and word tokenization.

[4] FOMC-text-preprocessing GitHub repository: https://github.com/rw19842/FOMC-text-preprocessing

Learning phrases

Originally my approach to implementing the n-gram creation rule was to simply calculate the frequencies of all n-grams with n ranging from 2 to 5. Then filter this list based on a threshold minimum frequency. Any occurrence of an n-gram in this filtered list could then be substituted by a single token with words separated by an underscore as opposed to white-space. However, this approach resulted in many unwanted n-grams being identified.

Normalized Pointwise Mutual Information. So instead I use Normalized Pointwise Mutual Information (NPMI) [7] to extract collocations from the text, that is, identify phrases or n-grams such as *New York* and *residential real estate.* A full list of n-grams can be found in appendix A. While NPMI did produce a higher quality list of n-grams, there is still room for improvement, in particular the list would benefit from a manual review including manual additions or removals. As in my previous approach, identified phrases are substituted by a single token.

NPMI gives a normalised measure of how much more likely two words are to occur together compared to their individual probabilities of occurring. The NPMI of a bigram, "$word_1\ word_2$" is calculated according to

$$\mathrm{NPMI}\left(word_1,\ word_2\right) = \frac{\mathrm{PMI}\left(word_1,\ word_2\right)}{-\log\left(p\left(word_1,\ word_2\right)\right)}, \tag{3.1}$$

where the Pointwise Mutual Information (PMI) is given by

$$\mathrm{PMI}\left(word_1,\ word_2\right) = \log\left(\frac{p\left(word_1,\ word_2\right)}{p\left(word_1\right)p\left(word_2\right)}\right). \tag{3.2}$$

In practice the probabilities in equations 3.1 and 3.2, $p\left(word_1,\ word_2\right),\ p\left(word_1\right),$ and $p\left(word_2\right)$, are approximated empirically from the text dataset by counting occurrences.

NPMI takes a value of -1 when the two words never appear together and 1 when the two only ever appear together. If the occurrence of the two words is independent the NPMI will be 0. The two tokens are considered collocations if their NPMI is above a chosen threshold. I make three passes of this with decreasing thresholds to identify phrases of more than two tokens. Initially, I set the threshold for each pass to be very low (0.4) and then examine the identified n-grams and their associated NPMI, rising the threshold to accordingly to filter out noise. The resulting thresholds are 0.66, 0.6, and 0.5. Additionally, I do not consider phrases whose frequency is less than 3,072, this speeds up training and reduces noise.

3.2.3 Preprocessing analysis

The purpose of implementing three progressively thorough configurations is to study the effects of text preprocessing and facilitate their fine-tuning to optimise performance of downstream NLP tasks. To evaluate each of the preprocessing configurations I assess the intrinsic quality of the subsequent text datasets. In particular, a high quality dataset is characterised by a smaller vocabulary size coupled with a relatively high average frequency of individual tokens. A preprocessing rule which decreases vocabulary size while increasing the average frequency, indicates that the rule successfully removed noise words which occur infrequently (with the exception of stopwords which are considered noise despite occurring frequently).

Table 3.2: Text Quality Statistics for each preprocessing configuration on the full dataset.

Configuration	# Tokens	Unique Tokens	Avg. Freq.	% Stopwords
Baseline	50,822,740	116,182	**437.44**	41.9%
Lightweight	28,664,680	**107,507**	266.63	0.0%
Heavyweight	**27,025,267**	107,679	250.98	0.0%

Comparing the `fomc_documents.csv` row of table 3.2 to B.1 we see that even the baseline configuration has a dramatic effect on the vocabulary size, reducing the number of unique tokens from 375,829 to 116,182. However, since the baseline doesn't remove stopwords there is in fact a negative impact on the proportion stopwords, rising from just under 31% to nearly 42%. Unsurprisingly, the lightweight configuration (which introduces the remove stopwords rule) significantly decreases the total number of tokens in the dataset to just shy of 28.7 million.

3.3 FOMC Word Embedding

3.3.1 Architecture

I use the Word2Vec algorithm described in section 2.5.2 for training a bespoke word embedding on the FOMC text dataset. Specifically, I use the skip-gram neural network architecture for its superior performance on smaller datasets.

Given the results of the preprocessing analysis 3.2.3, I opt to train my word embedding on the dataset preprocessed by the heavyweight configuration. I believe that the benefit of including relevant n-grams such as "trade deficit" and "euro area" in the vocabulary outweighs the marginal increase to the vocabulary size along with a slight decrease in average frequency.

3.3.2 Results

Creating a word embedding is an unsupervised task and so evaluating the quality of word vectors intrinsically is difficult – often they are evaluated against their performance in a downstream task, such as the word analogy task described in [17]. Hence, I instead carry out a sanity check by producing plots as in figure 3.1. For example, in figure 3.1 we find that words with similar meanings clump together (e.g. the names Powell, Yellen, Bernanke, Greenspan, and Volcker are the previous five chairpersons of the Federal Reserve). Likewise, "federal_fund_rate" is very close to "ffr" (its acronym) with respect to cosine similarity (defined in equation 2.1).

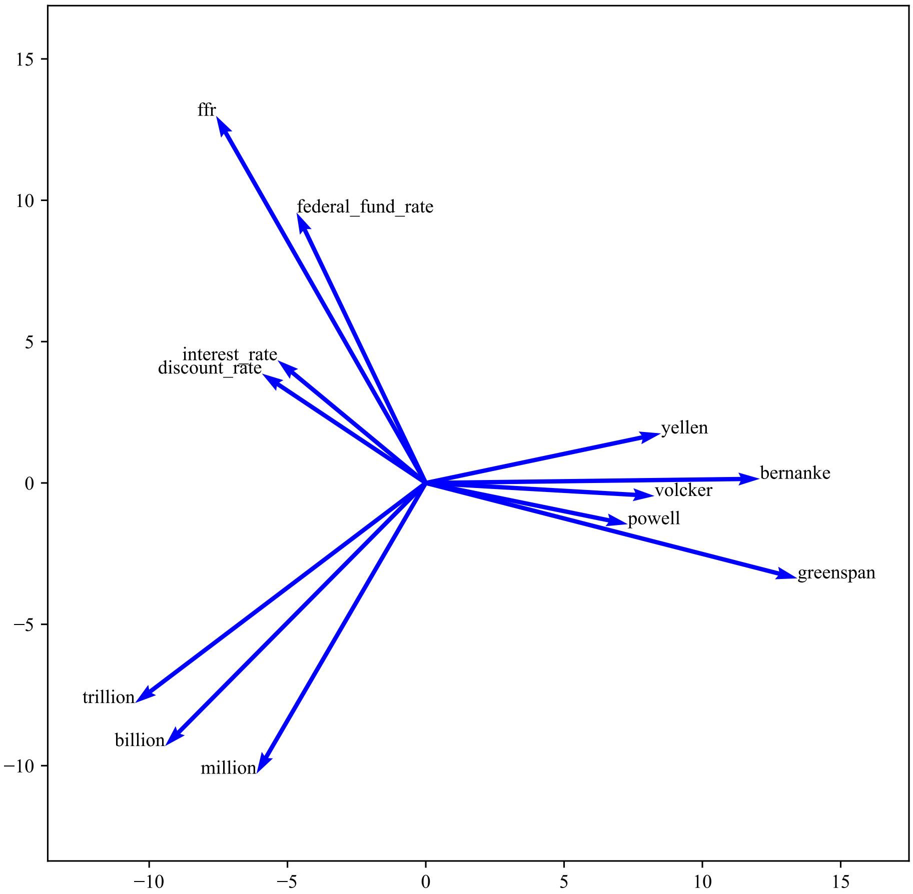

Figure 3.1: Visualization of bespoke word embedding using PCA to reduce dimensionality.

To depict the word vectors which have hundreds of dimensions in the two-dimensional figure I employ Principal Component Analysis (PCA) to reduce their dimensionality. More specifically, I first select the words I wish to plot and then apply PCA to the associated subset of word vectors. Finally, I plot the two most significant principal components of each word vector. This is implemented in a single Python function to streamline the process. However, this method necessarily results in a loss of information – the two dimensions depicted in figure 3.1 only explains 40.7% of the variance between the word vectors.

Chapter 4

Conclusion

In conclusion, management of monetary policy by central banks around the world, specifically through their control bank rates such as the United States' FFR, ripples through the banking system, ultimately shaping the financial landscape for everyday people. Whether it's through the cost of borrowing for mortgages, business loans, or other forms of credit, central bank actions significantly impact the spending power and financial well-being of individuals and businesses alike.

In this project I have compiled the most comprehensive dataset, to my knowledge, of text data surrounding monetary policy decisions in the United States. The dataset spanning almost a century contains over 62 million words over nearly 6,000 documents. In publishing my FOMC text dataset on Kaggle[1] and making the web scraping code available on GitHub[2] I hope to enable further research in applying NLP to the study of monetary policy. Likewise, I hope that my bespoke word embedding demonstrates the utility of modern NLP techniques and serves as inspiration for future work.

The introduction laid out numerous objectives in section 1.2, yet time constraints prevented their full pursuit. Namely, I had hoped to build a topic model which utilised my bespoke word vectors. With this topic model I could have achieved two things: evaluate my word embedding extrinsically compared to a pre-trained model such as the word vectors trained on 100 billion words from Google News [17, 16], and secondly I could have reproduced results from papers described in chapter 2. Additionally, without a topic model I could not have developed a trading strategy to demonstrate its utility.

In future, it would be interesting to see a wider application of NLP techniques to studying monetary policy outside the United States and to languages other than English. Likewise, exploration of applying alternative neural network architectures such as the Transformer introduced in [29] upon which OpenAI's ChatGPT is built.

[1] FOMC text dataset available at: www.kaggle.com/datasets/edwardbickerton/fomc-text-data

[2] Fed-Scraper GitHub repository: https://github.com/rw19842/Fed-Scraper

Bibliography

[1] Felipe Almeida and Geraldo Xexéo. *Word Embeddings: A Survey*. 2023. arXiv: 1901.09069 [cs.CL].

[2] Mikael Apel and Marianna Grimaldi. "The information content of central bank minutes". In: *Riksbank Research Paper Series* 92 (2012). URL: https://dx.doi.org/10.2139/ssrn.2092575.

[3] S Boragan Aruoba and Thomas Drechsel. "Identifying monetary policy shocks: A natural language approach". In: (2022). URL: https://repec.cepr.org/repec/cpr/ceprdp/DP17133.pdf.

[4] Yoshua Bengio et al. "A Neural Probabilistic Language Model". In: *Journal of Machine Learning Research* 3 (2003), pp. 1137–1155.

[5] BEN S. BERNANKE and KENNETH N. KUTTNER. "What Explains the Stock Market's Reaction to Federal Reserve Policy?" In: *The Journal of Finance* 60.3 (2005), pp. 1221–1257. URL: https://doi.org/10.1111/j.1540-6261.2005.00760.x.

[6] Steven Bird, Edward Loper, and Ewan Klein. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[7] Gerlof Bouma. "Normalized (pointwise) mutual information in collocation extraction". In: *Proceedings of GSCL* 30 (2009), pp. 31–40.

[8] Rob Churchill and Lisa Singh. "textPrep: A text preprocessing toolkit for topic modeling on social media data". In: *Proceedings of the 10th International Conference on Data Science, Technology and Applications*. 2021. URL: https://doi.org/10.5220/0010559000600070.

[9] Rob Churchill and Lisa Singh. "The Evolution of Topic Modeling". In: *ACM Comput. Surv.* 54.10s (Nov. 2022). ISSN: 0360-0300. DOI: 10.1145/3507900. URL: https://doi.org/10.1145/3507900.

[10] *FOMC Document Descriptions and Release Schedule*. Sept. 2023. URL: https://www.federalreserve.gov/monetarypolicy/fomc_historical.htm.

[11] Yuriy Gorodnichenko, Tho Pham, and Oleksandr Talavera. "The Voice of Monetary Policy". In: *American Economic Review* 113.2 (Feb. 2023), pp. 548–84. DOI: 10.1257/aer.20220129. URL: https://www.aeaweb.org/articles?id=10.1257/aer.20220129.

[12] Stephen Hansen, Michael McMahon, and Andrea Prat. "Transparency and Deliberation Within the FOMC: A Computational Linguistics Approach". In: *The Quarterly Journal of Economics* 133.2 (2017), pp. 801–870. ISSN: 0033-5533. DOI: 10.1093/qje/qjx045. URL: https://doi.org/10.1093/qje/qjx045.

[13] Tarek Alexander Hassan et al. "The Global Impact of Brexit Uncertainty". In: Working Paper Series 26609 (Jan. 2020). DOI: 10.3386/w26609. URL: https://dx.doi.org/10.2139/ssrn.3840164.

[14] John S Justeson and Slava M Katz. "Technical terminology: some linguistic properties and an algorithm for identification in text". In: *Natural language engineering* 1.1 (1995), pp. 9–27. URL: https://doi.org/10.1017/S1351324900000048.

[15] Tim Loughran and Bill McDonald. "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks". In: *The Journal of finance* 66.1 (2011), pp. 35–65. URL: https://doi.org/10.1111/j.1540-6261.2010.01625.x.

[16] Tomas Mikolov et al. *Distributed Representations of Words and Phrases and their Compositionality*. 2013. arXiv: 1310.4546 [cs.CL].

[17] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[18] Frederic Morin and Yoshua Bengio. "Hierarchical Probabilistic Neural Network Language Model". In: *International Conference on Artificial Intelligence and Statistics*. 2005. URL: https://api.semanticscholar.org/CorpusID:1326925.

[19] Andreas Neuhierl and Michael Weber. "Monetary policy communication, policy slope, and the stock market". In: *Journal of Monetary Economics* 108 (2019), pp. 140–155. ISSN: 0304-3932. URL: https://doi.org/10.1016/j.jmoneco.2019.08.005.

[20] Dat Quoc Nguyen et al. "Improving Topic Models with Latent Feature Word Representations". In: *Transactions of the Association for Computational Linguistics* 3 (June 2015), pp. 299–313. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00140. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00140/1566790/tacl_a_00140.pdf. URL: https://doi.org/10.1162/tacl%5C_a%5C_00140.

[21] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[22] Zahra Rahimi and Mohammad Mehdi Homayounpour. "The impact of preprocessing on word embedding quality: a comparative study". In: *Language Resources and Evaluation* 57.1 (2023), pp. 257–291. DOI: 10.1007/s10579-022-09620-5. URL: https://doi.org/10.1007/s10579-022-09620-5.

[23] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[24] Christina D. Romer and David H. Romer. "A New Measure of Monetary Shocks: Derivation and Implications". In: *American Economic Review* 94.4 (Sept. 2004), pp. 1055–1084. DOI: 10.1257/0002828042002651. URL: https://www.aeaweb.org/articles?id=10.1257/0002828042002651.

[25] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0. URL: https://doi.org/10.1038/323533a0.

[26] Agam Shah, Suvan Paturi, and Sudheer Chava. *Trillion Dollar Words: A New Financial Dataset, Task & Market Analysis*. 2023. URL: https://arxiv.org/abs/2305.07972.

[27] Christopher A. Sims. "Interpreting the macroeconomic time series facts: The effects of monetary policy". In: *European Economic Review* 36.5 (1992), pp. 975–1000. ISSN: 0014-2921. DOI: https://doi.org/10.1016/0014-2921(92)90041-T. URL: https://www.sciencedirect.com/science/article/pii/001429219290041T.

[28] Kristina Toutanova et al. "Feature-rich part-of-speech tagging with a cyclic dependency network". In: *Proceedings of the 2003 human language technology conference of the north american chapter of the association for computational linguistics*. 2003, pp. 252–259. URL: https://aclanthology.org/N03-1033.

[29] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[30] Felipe Viegas et al. "CluHTM - Semantic Hierarchical Topic Modeling based on CluWords". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020. URL: https://aclanthology.org/2020.acl-main.724.

[31] Shirui Wang, Wenan Zhou, and Chao Jiang. "A survey of word embeddings based on deep learning". In: *Computing* 102.3 (2020), pp. 717–740.

[32] Zyte. *Scrapy: An open source and collaborative framework for extracting the data you need from websites. In a fast, simple, yet extensible way.* 2023. URL: https://scrapy.org.

Appendix A

Phrases

Table A.1: Identified phrases using the collocation statistics of `fomc_documents.csv`

	Phrase	Frequency
1	interest rate	54,239
2	federal fund rate	32,527
3	monetary policy	32,283
4	unemployment rate	29,558
5	fourth quarter	27,088
6	percentage point	24,119
7	first quarter	23,654
8	federal reserve bank	23,558
9	last year	22,723
10	basis point	22,344
11	open market committee	21,173
12	federal open market	20,417
13	second quarter	19,927
14	economic activity	19,807
15	money market	18,427
16	third quarter	18,056
17	real gdp	18,024
18	chairman greenspan	17,367
19	chairman volcker	14,982
20	billion dollar	14,358
21	intermeeting period	13,115
22	discount rate	13,097
23	united state	13,088
24	commercial bank	12,998
25	retail sale	12,664
26	vice chairman	12,556
27	industrial production	12,530
28	first half	12,502
29	inflation expectation	12,095
30	year ago	11,858
31	consumer spending	11,779
32	second half	11,518
33	balance sheet	11,188
34	since last	10,541
35	central bank	10,267
36	treasury security	9,996
37	bank new york	9,728

Continued on next page

Table A.1: Identified phrases using the collocation statistics of `fomc_documents.csv`

	Phrase	Frequency
38	good service	9,607
39	monetary aggregate	9,416
40	previous tealbook	9,127
41	current account	9,077
42	motor vehicle	9,012
43	open market account	8,947
44	foreign currency	8,876
45	financial condition	8,774
46	recent week	8,621
47	net export	8,557
48	chairman bernanke	8,546
49	treasury bill	8,517
50	price stability	8,421
51	rate per cent	8,220
52	government security	8,184
53	january february	8,060
54	federal reserve system	7,990
55	open market operation	7,940
56	money supply	7,935
57	new order	7,763
58	july august	7,709
59	food energy	7,661
60	labor force	7,199
61	foreign exchange	7,111
62	thank chairman	7,099
63	san francisco	7,026
64	balance payment	6,983
65	october november	6,966
66	commercial real estate	6,945
67	commercial paper	6,902
68	june july	6,741
69	capital good	6,676
70	board governor federal reserve	6,617
71	personal income	6,482
72	housing start	6,420
73	fomc meeting	6,394
74	million unit	6,348
75	durable good	6,318
76	core pce	6,291
77	target range	6,267
78	november december	6,260
79	february march	6,235
80	chairman burn	6,217
81	labor cost	6,007
82	fix investment	6,001
83	six month	5,999
84	near term	5,841
85	class fomc	5,800
86	capital spending	5,613
87	asset purchase	5,415
88	except noted	5,348
89	state local government	5,261
90	even though	5,209

Continued on next page

Table A.1: Identified phrases using the collocation statistics of `fomc_documents.csv`

	Phrase	Frequency
91	payroll employment	5,181
92	core inflation	5,144
93	saving deposit	5,102
94	real gnp	5,082
95	labor market condition	5,055
96	united kingdom	5,031
97	seasonally adjust annual rate	4,982
98	upward pressure	4,867
99	corporate bond	4,835
100	unanimous vote	4,821
101	downside risk	4,813
102	chairman martin	4,677
103	coming month	4,638
104	executive committee	4,544
105	inflationary pressure	4,386
106	export import	4,356
107	industrial country	4,239
108	long run	4,237
109	domestic nonfinancial	4,227
110	vice president federal reserve	4,215
111	committee seek	4,191
112	capacity utilization	4,118
113	thrift institution	4,055
114	reserve requirement	3,983
115	maximum employment	3,964
116	bond yield	3,961
117	kansas city	3,901
118	agency mortgagebacked security	3,900
119	natural gas	3,896
120	research statistic	3,881
121	per hour	3,795
122	free reserve	3,787
123	incoming data	3,785
124	downward pressure	3,715
125	repurchase agreement	3,646
126	division research	3,624
127	mutual fund	3,579
128	nonfarm payroll	3,572
129	euro area	3,553
130	assistant secretary	3,505
131	director division	3,501
132	personal consumption	3,479
133	per barrel	3,477
134	yield curve	3,469
135	output gap	3,462
136	financial institution	3,459
137	commercial industrial	3,451
138	august sept	3,443
139	gross national product	3,375
140	unit labor	3,349
141	vote action	3,344
142	excess reserve	3,299
143	trade deficit	3,290

Continued on next page

Table A.1: Identified phrases using the collocation statistics of `fomc_documents.csv`

	Phrase	Frequency
144	past three	3,279
145	rise per cent	3,275
146	third district	3,250
147	nonresidential construction	3,247
148	chair yellen	3,242
149	agency debt	3,237
150	please note	3,179
151	general counsel	3,133
152	manager system	3,127
153	econ devel	3,104
154	residential real estate	3,067

Appendix B

Dataset File Descriptions

Table B.1: Statistics of raw tokenised text dataset.

File	# Docs.	# Tokens	Tokens/Doc.	% Stopwords	Unique Tokens	Avg. Freq.
fomc_documents.csv	5,958	62,846,088	10,548.19	30.9%	375,829	167.22
miscellaneous.csv	614	3,075,209	5,008.48	29.3%	44,692	68.81
meeting_minutes.csv	1,799	12,666,209	7,040.69	36.4%	52,804	239.87
bluebooks.csv	485	4,537,943	9,356.58	22.2%	101,155	44.86
agendas.csv	588	146,053	248.39	21.9%	2,469	59.15
meeting_transcripts.csv	461	17,200,800	37,311.93	39.2%	81,846	210.16
press_conference_transcript.csv	67	432,435	6,454.25	39.1%	12,601	34.32
greenbooks.csv	1,260	18,573,953	14,741.23	22.1%	217,027	85.58
redbooks.csv	465	6,108,935	13,137.49	30.0%	42,420	144.01
policy_statements.csv	219	104,551	477.40	29.5%	2,096	49.88

B.1 fomc_documents.csv

The fomc_documents.csv file is a superset of the others, containing all the documents scraped from the Federal Reserve website.

B.2 miscellaneous.csv

Any documents which are not sorted into one of the below files is placed in miscellaneous.csv. For example presentation_materials and accessible_materials are both document kinds found in this file.

B.3 meeting_minutes.csv

The meeting_minutes.csv file contains the following document kinds: record_of_policy_action, historical_minutes, minutes_of_actions, minutes, intermeeting_executive_committee_minutes, and memoranda_of_discussion. Each of which gives a summary of the meetings, albeit with varying detail and release schedules. For further details on each document kind see [10].

B.4 bluebooks.csv

The bluebooks.csv file contains two document kinds: bluebook and tealbook_b. First created in 1965 the Bluebook went through a series of name changes, originally titled “Money Market and Reserve Relationships” in 1970 it was renamed to “Monetary Aggregates and Money Market Conditions” and finally in 1981 to “Monetary Policy Alternatives”. In June 2010 the Tealbook (officially titled “Report to the FOMC on Economic Conditions and Monetary Policy") was created when the Bluebook was merged with the Greenbook. Tealbook B, officially titled "Monetary Policy: Strategies and Alternatives", includes the same information as the deprecated Bluebook.

B.5 agendas.csv

As the name suggests agendas.csv contains documents of kind agenda which each contain a list of items to be covered at each FOMC meeting, given to each committee member a week before the meeting.

B.6 meeting_transcripts.csv

The meeting transcripts are the most detailed record of FOMC meetings, created from audio recordings of the meetings with only minimal editing. The file meeting_transcripts.csv contains transcripts (document kind transcript) from 1976 to 2017. Prior to 1993 FOMC meetings were tape recorded for the sole purpose of preparing minutes; unbeknownst to members, transcripts were also created and archived. In 1993 all previous transcripts were released and since then transcripts have been released with a 5-year lag.

B.7 press_conference_transcript.csv

In 2011 the Federal Reserve began holding a press conference shortly after each FOMC meeting in which the policy decision is announced along with context and justification. There is also a chance for press representatives to question the chairman. As implied by the filename, meeting_transcripts.csv contains the transcripts of these press conferences.

B.8 greenbooks.csv

File greenbooks.csv contains the document kinds: greenbook, greenbook_supplement, greenbook_part_one, greenbook_part_two, and tealbook_a. Officially titled "Current Economic and Financial Conditions", the Greenbook was given to FOMC members one week before each meeting from 1964 to 2010. Its successor, Tealbook A (officially titled "Economic and Financial Conditions: Current Situation and Outlook") provides in-depth analysis and forecasts of both the U.S. and international economy previously covered in the Greenbook in addition to analysis of financial markets previously found in the Bluebook.

B.9 redbooks.csv

Officially titled "Current Economic Comment by District", the discontinued Redbooks were internal documents produced from 1970 until 1983 when its content was reformulated into the Beige Book (officially titled "Summary of Commentary on Current Economic Conditions by Federal Reserve District"). The main difference between the Beige Book and its precursor is that it gets released *ahead* of each scheduled FOMC meeting. As such, redbooks.csv contains documents of both kinds, beige_book and redbook.

B.10 policy_statements.csv

Finally, file policy_statements.csv contains documents of kind statement and implementation_note. These statements are released to the public following each meeting and disclose the policy decisions made by the committee. The FOMC only began doing this in 1994.