Imperial College London

Imperial College London

Department of Mathematics

An Automated Approach On Generating Macro-Economic Sentiment Index Through Central Bank Documents

Author:
Ruizhe Xia (CID: 01201752)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2019-2020

Abstract

I proposed an automated approach on creating region-specific dictionary to quantify central bank's view on macroeconomics and interest rate using documents including press conference scripts, monetary policy committee meeting minutes, member's speeches and statements, which outperforms any other existing dictionary and even the most recent FinBERT model [1] which contains more than 110 million parameters. This thesis introduced an innovative way to label the document with positive/negative/neutral sentiment in order to construct a new economical dictionary. This thesis demonstrated how the Latent Dirichlet Allocation[1][2] can be applied as a language noise filter to remove the less relevant sentences. And this paper compared the dictionary performance with two famous economic dictionaries of 'Financial Stability Dictionary' and 'Loughran McDonald Dictionary', and proved that self-created dictionary is having better and consistent performance than the existing dictionary. This paper also showed how the dictionary can work as a feature reduction method for working with Machine Learning Models when the availability of documents is limited.

The goal is not only just to provide a quantitative way to interpret the economical meaning of central banks documents, but also to show how the central bank sentiment is closely related to LIBOR rates. In this paper, I will primarily demonstrate how the algorithm is applied to Bank of England materials, but the model is largely applicable on all other major central banks, namely ECB and FED. In the fourth chapter of this thesis, I will show hows the sentiment looks like on each central banks and its local rates.

The sentiment performance is also fairly promising. I trained the dictionary only on documents before 2009 to prevent looking forward bias. The final dictionary consists of only 293 words with 124 positive words and 169 negative words. And this dictionary still captures interest rate sentiment until 2020, by achieving 0.351 correlation with Sterling LIBOR 1Y rate. The performance is further validated by using the model train from BOE onto other central banks' documents, namely ECB and FED, where it still shows a correlation of 0.336 with their respective local LIBOR rates, comparing to 0.255 correlation for FS dictionary sentiment and 0.04 correlation on LM dictionary sentiment.

To further validate the performance of the model, I introduced the most recent financial NLP model, Finbert [1], and proved that complicated models not necessarily outperform the simple model. The simple dictionary-based sentiment based on filtered speech and minutes shows a better correlation with the LIBOR rate than the Finbert sentiment. And finally, I introduce a new method to measure the predictive power of various sentiment on various economical indices, including GDP, CPI, unemployment rate and interest rate, which validates the strong connections between central bank documents' sentiment and macro-economic conditions.

[1]Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John (ed.). "Latent Dirichlet Allocation". Journal of Machine Learning Research. 3 (4–5): pp. 993–1022. doi:10.1162/jmlr.2003.3.4-5.993. Archived from the original on 2012-05-01. Retrieved 2006-12-19.

Acknowledgements

I need to appreciate Cissy Chan and many other people from BlackRock for guiding me through this project. They did offer me lots of great insights and suggestion on text mining and the understanding of central bank policies and languages. This thesis would not be possible without their patience and assistance in the past few months.

And I also need to appreciate my supervisor Johannes Muhle-Karbe for the suggestions on modifying thesis structures and on improving the explanations of various algorithms I implemented throughout the project.

And finally, I need to thank Stack Overflow, Wikipedia and Google for helping me debug my code and scripts.

Contents

Chapter 1

Introduction

1.1 Literature Review

Sentiment analysis is a model that aims to extract sentiment, opinion or attitudes of people/institution from written language, including but not limited to sentences, paragraphs, documents, social media. [3]. And normally there are two approaches regarding this task: (1) machine algorithms that extract features and sentiments from texts with 'word count' representation [4, 5, 6, 7], (2) deep learning-based algorithms that extract sentiments from texts which are represented by different types of embedding[8, 9]. The first methods are easy to implement but suffer from its failure of ignoring semantic information based on the order of sequence or dependency between different vocabularies. The second method is often required a huge amount of labelled text data of the same context to learn its explosively large number of parameters.

In Natural Language Processing **(NLP)** community, there has been numerous successful development and wide-spread discussion on language model and text classification, but it was majorly focused on conversational language rather than financial and economical language. And the majority of successful NLP models were trained on a huge amount of text data, which is highly impossible in our scenarios. (At least hundreds of thousands labelled sentences on economical content to train the over 100 million parameters in different NLP model) Take one example of the very recent successful language model was the "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" published by researchers at Google AI Language, which was pre-trained with over 110 million parameters. Contrasting to the popularity and fast evolutions of generic NLP models (BERT 2018, XLNET 2019, GPT3 2020), researches and developments in language model on financial texts, especially on macro-economics topics, seems to be stagnated and less popular. The main reason that to pre-train an accurate Bert model would require millions of economical documents which is generally not available for most of researchers.

One of the very first language model on financial context was developed by Loughran and McDonald (2011) "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks"[7] in understanding the tones of 10-Ks, suggesting that the contextual sentiments of same words could be significantly different in different text sources. And further proven by the 2017 paper "Sentiment in Central Banks' Financial Stability Reports"[6], they suggest the financial stability dictionary based on Federal reserve communication is 30%different to dictionary created by Correa, Ricardo, Keshav Garud, Juan M. Londono, and Nathan Mislang (2017) which is based on US public listed companies' annual reports. Hence, the language model should need to be re-trained based on the specific context before any forecasting/classification purposes.

There have been several papers talking about central banks communication and deriving sentiments, but most of the papers were using the news data-set, such as Dow Jones Newswires[10] and Factiva dataset[11], to look at central bank communication rather than looking at the central bank documents directly. And most of the paper simply using the existing dictionary rather than creating task-specific dictionary. The 2015 paper "Does Central Bank Tone Move Asset Prices?"[12] looked at ECB press conferences held after ECB policy meeting. And they proved the changes in ECB tine convey generic information for stock markets as well as views of future

monetary policy. The May 2020 Paper, "Making text count: economic forecasting using newspaper text"[13], had used 10 different dictionary-based sentiment indexes for forecasting macroeconomics movement, including Financial Stability dictionary, Loughran and McDonald dictionary, social media sentiment dictionary[1], Harvard IV psychological dictionary, Anxiety and Excitement dictionary[2], Economic Uncertainty[3], Monetary Policy Uncertainty[4] and Economic Policy Uncertainty dictionary[5]. And the financial stability dictionary has been the best-performed dictionary [13] correlated with various indicators including "OECD UK business confidence", "Lloyds Business Barometer" and "Composite PMI", in term of size of error between predicted and actual financial indicators. In my research, I will focus on comparing the performance of Financial stability dictionary as my baseline model.

There has also been limited research on the sentiment of central bank minutes, speech and statements. We would expect that the communication of central banks reveals their opinions on short-term and long-term macroeconomic movements, such as the interest rate movement and LIBOR rates. The 2019 paper of "Narrative monetary policy surprises and the media" [14] reveals a significant information-surprise on various topics after the central bank documents being published, including interest rate, risk, uncertainty and growth. Both "Measuring Central Bank Communication: An Automated Approach with Application to FOMC Statements"[15] and "Between hawks and doves: measuring central bank communication "[16] and proved significant relation between central bank document sentiment index and treasury bills returns and interest rates. Hence, it is theoretically feasible to generate sentiment index based on central bank documents, while greatly reflects banks' opinions and predicts on economical and financial indicators' movements.

1.2 Project Overview

Natural language processing on central bank documents has always been a challenge because of the limited documents available. The popular deep learning model such as XLNET and BERT requires at least hundreds of thousands of labelled document which is very unlikely to be available. It is because all the latest NLP language models are having hundreds of millions of parameters[6]. (And in this thesis, I will also show that my model outperforms the most recent FinBERT model [1] on interest rate prediction tasks.) Bank of England (BOE) usually post one minute per month and 3 speech per month. This means I was working with only 1044 documents starting from 1997 to 2020 to train a decent language model. This is also the reason why there are limited researches on the central bank minutes and speeches directly, and why most of the financial NLP papers are working with news database or cooperate finance documents. But this also makes me curious about how much information can be extracted from the central bank documents and on how well it will perform on explaining and forecasting the movement of various financial indexes. (eg: LIBOR, GDP, Unemployment rate)

In this project, the main motivation is to understand how interest rate sentiment can be extracted from central bank communication documents and how well the language model could perform despite the constraints on data availability. I decided to take the challenge to automate a sentiment generation process to quantify the central bank's view on the macroeconomic situation and establish a link to existing financial and economical indexes. Because of limited data availability, I intentionally decided to take some simple model on extracting sentiments out of BOE minutes and speeches, such as dictionary-based model and Multinomial Naive Bayesian model, instead of a deep-learning-based model.

[1]Finn Årup Nielsen, 2011, 'A new ANEW: Evaluation of a word list for sentiment analysis in microblogs', https://arxiv.org/abs/1103.2903

[2]Rickard Nyman, Sujit Kapadia, David Tuckett, David Gregory, Paul Ormerod and Robert Smith, 'News and narratives in financial systems: exploiting big data for systemic risk assessment', https://www.bankofengland.co.uk/working-paper/2018/news-and-narratives-in-financial-systems

[3]Michelle Alexopoulos Jon Cohen, 2009. "Uncertain Times, uncertain measures," Working Papers tecipa-352, University of Toronto, Department of Economics

[4]Husted, Lucas, John Rogers, and Bo Sun (2017). "Monetary Policy Uncertainty". International Finance Discussion Papers 1215. https://doi.org/10.17016/IFDP.2017.1215

[5]Scott R. Baker, Nicholas Bloom, Steven J. Davis, Measuring Economic Policy Uncertainty, The Quarterly Journal of Economics, Volume 131, Issue 4, November 2016, Pages 1593–1636

[6]BERT base has 12 layers (transformer blocks), 12 attention heads, and 110 million parameters. BERT Large has 24 layers, 16 attention heads and, 340 million parameters. XLNet has 24-layer, 1024-hidden, 16-heads, 340M parameters. GPT-2 has 1.5 billion parameters

And Latent Dirichlet allocation (LDA)[2] has been a famous and successful probabilistic method in discover topics from a set of documents [17]. LDA assumes each document has a probability distribution over different topics, and each topic has a probability distribution over different vocabularies. So based on the words that occurred in the text, the LDA model could give a probability distribution over different topics for this text (either sentences, paragraphs or documents). You can understand more about LDA at Section 3.1. But in this thesis, I aim to use LDA for two purposes: (1) cluster paragraphs in minutes into 6 different topics (2) semantically clean up the language of speeches by removing semantic incoherent sentences in the speech.

This thesis contributes to macro-economic sentiment construction consist of two parts:

1. By learning a new interest-rate dictionary based on the Bank of England documents, the dictionary-based sentiment that constructed on BOE minutes and speeches can capture the bank's opinion and sentiment on the current macro-economic level. And the sentiment constructed is having a strong connection with UK based rate and LIBOR rate movement. But the same methodology can easily be applied to other banks. I will also demonstrate how well the sentiment constructed on FED and ECB documents are closely connected to their local LIBOR rate, in addition to BOE documents.

2. Innovative using Latent Dirichlet allocation (LDA) (see Section 3.3 for more details) as language style filter to remove less relevant sentences in BOE speech and only keep relevant sentences that have similar language style as BOE minutes. This semantic language cleaning significantly improved the dictionary-based model performance.

To construct the sentiment index on UK, I first build up web scraper on BOE websites to download all the 1044 files available. Then I use the document from 1997-2009 as the training data, the documents starting 2010 are viewed as testing data (out-of-sample documents) to validate the consistency of performance. The label of the document is based on if there is a local risk rise or drop within the next 6 months after the document published. In this case, if UK base rate increases within the next 6 month, the document is label as positive sentiment document. If UK base rate drops within the next 6 month, the document is labelled as negative sentiment document. And if neither things happen, the document is then labelled as neutral sentiment document.

Then I assume the work with positive sentiment will appear more frequently in positive sentiment documents and vice versa. The dictionary was then selected based on the possibility of document being positive/negative given this word appeared. After the dictionary was trained, the sentiment of each document was calculated by looking at the difference between positive and negative scores divided by the sum of the positive and negative score. And the UK interest rate sentiment was constructed by averaging all texts published in the same month, and exponentially moving averaged with past 3 months interest rate sentiment. And the final dictionary-based sentiment is classified as interest rate sentiment in the next month. I quantify the sentiment performance by looking at the correlation between sentiments and Sterling LIBOR 1Y rate.

But I found the dictionary-based method applied on minutes was consistent, but performance on speech was horribly performed with 0.3 correlation on training data but 0 correlation on testing data. Hence, I innovative used Latent Dirichlet allocation (LDA) to remove the noise in BOE speech and the performance improve significantly. And the final dictionary size is 294 words with 124 positive words and 169 negative words. And the dictionary performance is significantly better than sentiment constructed based on 'Financial Stability Dictionary' and 'Loughran McDonald Dictionary'.

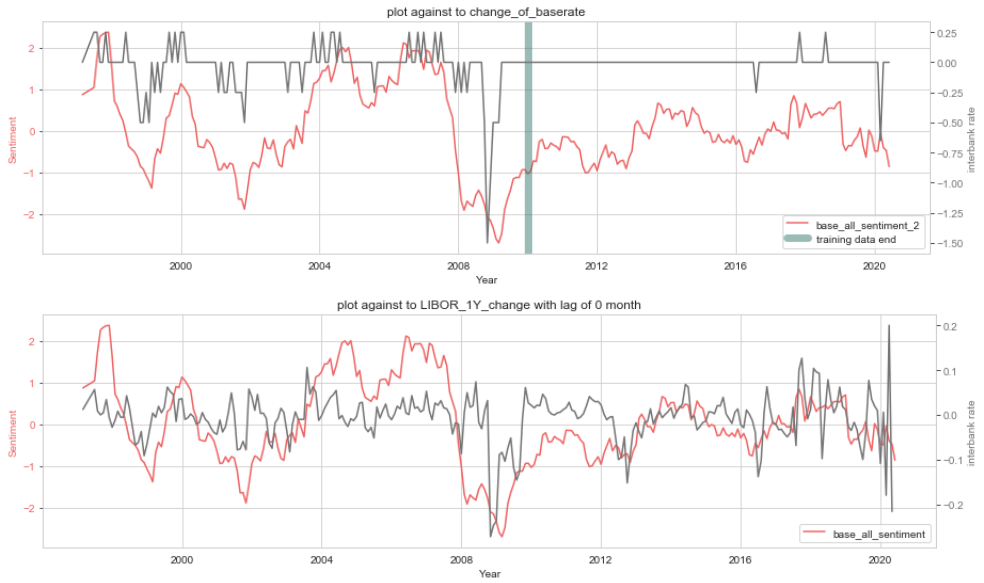

Figure 1.1: Sentiment index created by the final model and plotted against UK LIBOR 1Y rate and base rate

In addition, I talked about two model extensions. I first went one step forward to build work embedding on these 294 words and constructed a multi-nominal Naive Bayesian model to better indicate possible base rate movements. But the main take away is that the dictionary construction method proposed in this thesis can act as a feature selection algorithm for different NLP tasks. And I secondly applied the same methodology on FED and ECB documents and achieved still achieve high correlation with those local interest rate. This validates the economical meaning and performance of the model even further.

And finally, I introduced the most recent and powerful pre-trained model FinBERT and compared its performance with my UK interest rate dictionary sentiment. And I showed that my model actually performed better, in term of both speed and accuracy, than the FinBERT model that has 110 million parameters. And I also introduced new metrics rather than correlation to validate that my model carries the highest amount of new information on interest rate prediction tasks than many other algorithms and existing indexes.

1.3 Data

All data used in this thesis are all publicly available. Those data are either downloaded directly from the web-page or download by the web-scraper developed by myself. But I will not discuss how the web-scraper was constructed in this thesis because this is not very relevant. I only downloaded minutes and speeches from central bank official website and ignoring other documents at the current stage. And the documents downloaded are unlabelled, and hence adding an automated and accurate document label is an essential step in this project.

For more details of the data sources:

1. Bank of England speech:
   https://www.bankofengland.co.uk/news/speeches
2. Bank of England minute:
   https://www.bankofengland.co.uk/news/news
3. The Federal Reserve minutes:
   https://www.fedsearch.org/fomc-docs/search?advanced_search=true&from_year=1997&search_precision=All+Words&start=0&sort=Relevance&to_month=7&to_year=2020&number=10&fomc_document_type=minutes&Search=Search&text=&from_month=3
4. European Central Bank Press Conference:
   https://www.ecb.europa.eu/press/pressconf/html/index.en.html
5. European Central Bank Monetary policy accounts:
   https://www.ecb.europa.eu/press/accounts/html/index.en.html
6. LIBOR 1Y 6M 3M 1M rate:
   https://fred.stlouisfed.org/series/EUR12MD156N
7. UK base rate, FED base rate, ECB base rate are easily found on their respective central bank website

And here is the summary on the text data I am using in this project

	Number of document	**Word Count after cleaning**	**Date range**
Bank of England minute	783	2200	1997-2020
Bank of England speech	261	2080	1997-2020
The Federal Reserve minute	324	3868	1997-2020
ECB policy accounts	45	3644	2015-2020
ECB policy press conference	240	837	1998-2020

Table 1.1: Descriptive statistics of downloaded documents from 4 different major central banks.

Noted that the table above only includes the articles after I removed all the non-relevant, repeated documents. So if you build the web-scraper using the address above, you may download a much higher amount of documents that the number stated here. I have applied a filter on the title to make sure this document is either shared by the Monetary Policy Committee or explaining the reasons behind the macro-economic policy movement.

Chapter 2

Dictionary Based Model (Baseline)

For the most of the time, the evaluation of text documents is through the subjective measure. From the textbook articles we read in our school time, to the news articles, economics documents or even academic papers, people's understanding of those articles could be different from each other. And when people read the same articles at different time and occasions, they might have a slightly different understanding of the sentiments and the implications in the text documents. And the evaluation of the sentiments in the text documents can be time-consuming as well as being inconsistent. Using an automated approach, such as sentiment dictionary on specific contents (such as economical, political or conversational), could give a more objective evaluation of the document, and we could perform a direct comparison on sentiments between documents published at different times from different authors. And such sentiment constructed on documents can remove the inconsistencies from the viewer's personal opinion and experiences. A sentiment index is only meaningful when the sentiment is generated from the consistent and objective evaluation.

In this section, I will first talk about the existing financial or economical dictionary, explain how to construct document sentiment based on these two dictionaries as well as quantify their performances. After that, I will introduce the method to generate document labels and to constructed a new region-specific economical dictionary. And final I will compare the performance of this newly created dictionary with existing measures.

In term of the document I was working with, I have downloaded files from the website of Bank of England (BOE), The Federal Reserve (FED) and European Central Bank (ECB)from 1997 to 2020. There are 783 BOE Speech, 261 BOE minutes, 285 ECB minutes, and 324 FED minutes. I will skip the details of web scraping in this paper.

But before the document can be used for any quantitative algorithm, we need to pre-process the document through the process of:

1. Remove documents that were too short, which carries limited information and would create bias in the sentiments. (Less than 100 words)
2. Remove punctuation, hyperlinks, references, footnotes, HTML tags, special non-Unicode characters, numbers and mathematical equations
3. Set all characters into lower cases
4. Remove the phrases and convert each word into is the base form, in another word, stemming and lemmatization[1]
5. Remove all the stop words (e.g. for, very, and, of, are, etc) which was chosen from famous NLTK word list proposed by Bird and Loper in 2004[2].

[1]Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. For more details, please refer to https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

[2]Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit, Steven Bird, Ewan Klein, and Edward Loper, 2004, https://www.nltk.org/book/

Below shows the example of documents before and after the pre-processing.

Original Document:

The Bank of England's Monetary Policy Committee (MPC) sets monetary policy to meet the 2% inflation target, and in a way that helps to sustain growth and employment. At its meeting ending 3 August 2016, the MPC voted for a package of measures designed to provide additional support to growth and to achieve a sustainable return of inflation to the target. This package comprises: a 25 basis point cut in Bank Rate to 0.25%; a new Term Funding Scheme to reinforce the pass-through of the cut in Bank Rate; the purchase of up to £10 billion of UK corporate bonds; and an expansion of the asset purchase scheme for UK government bonds of £60 billion, taking the total stock of these asset purchases to £435 billion. The last three elements will be financed by the issuance of central bank reserves.

After language processing:

bank england monetari polici committe mpc set monetari polici meet inflat target way help sustain growth employ meet end august mpc vote packag measur design provid addit support growth achiev sustain return inflat target packag compris basi point cut bank rate new term fund scheme reinforc pas cut bank rate purchas billion uk corpor bond expans asset purchas scheme uk govern bond billion take total stock asset purchas billion last three element financ issuanc central bank reserv

Figure 2.1: Document before and after preprocessing

2.1 Existing Dictionary

word	sentiment
Abandon	Negative
Abdicating	Negative
Abnormal	Negative
Able	Positive
Abundance	Positive
Acclaimed	Positive

Table 2.1: LM Dictionary

word	sentiment
Able	Positive
Abnormally	Negative
Abrupt	Negative
Absorb	Positive
Absorbed	Positive
Absorbing	Positive

Table 2.2: FS Dictionary

There are two existing dictionaries I will discuss: Loughran and McDonald 2011, and Financial Stability dictionary. Both of the dictionaries classify some word as positive sentiment words and some words as negative sentiment words. LM dictionary consists of 229 words and was trained on 50115 pieces of listed firms annual reports from 1994 to 2008. The FS dictionary consists of 391 words and was trained on 982 financial stability reports of 62 countries, ECB, and IMF published between 2000 and 2015.

To calculate the sentiment of a document, we need to first convert vocabulary in both dictionaries to normalised form by **Stemming and Lemmatization**. For example, the words from LM dictionary "ABANDON", "ABANDONED", "ABANDONING", "ABANDONMENT", "ABANDONMENTS", "ABANDONS" are all converted to the same word "abandon". Then the document sentiment based on LM or FS dictionary is simply:

$$\text{Sentiment of Document} = \frac{\#P - \#N}{\#P + \#N}$$

where $\#P$ represents the sum of the frequencies of all positive vocabularies from LM/FS dictionary that showed up in the document and $\#N$ represents the sum of the frequencies of all negative vocabularies showed in the document. Hence words were neither negative nor positive are ignored. And there is no difference between negative words and very negative words. (This drawback will be solved in the next section)

The final monthly central bank sentiment index is constructed by first averaging all texts published in the same month (eg: May). Then it is exponentially moving averaged with half-life equalling 3 month[3]. The final sentiment is classified as the sentiment for next month (eg: June)

If I apply the above methods based on LM dictionary or FS dictionary on 1044 Bank of England documents, I create LM sentiment and FS sentiment respectively. According to the method I set up the document label, the sentiment should optimally go down when the central bank has a more pessimistic view on the macro-economics, and the sentiment should optimally go up when the central bank has a more optimistic view on the macro-economics. And the optimistic/pessimistic view can be validated by observing either a rate cut or a rate hike in the next few months to control inflation or to boost the economy.

[3]The choice of 3 months was based on maximising the constructed sentiment's correlation with LIBOR 1Y monthly change

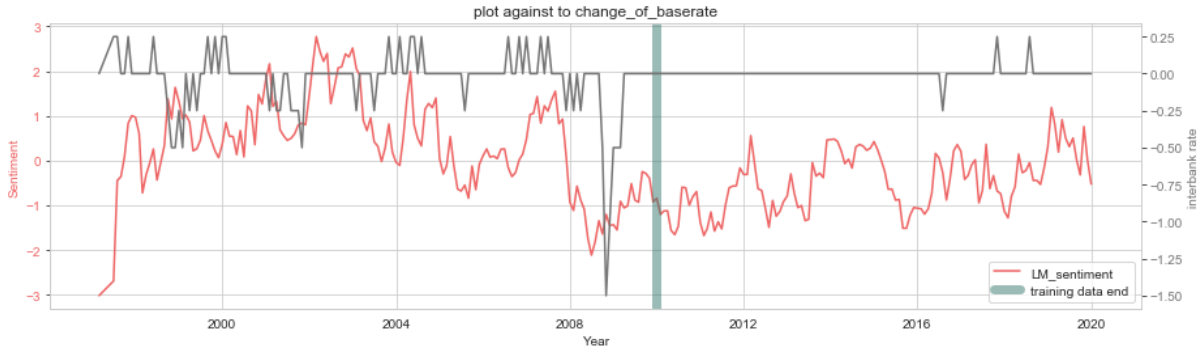

Figure 2.2: LM dictionary sentiment on BOE documents and UK base rate change

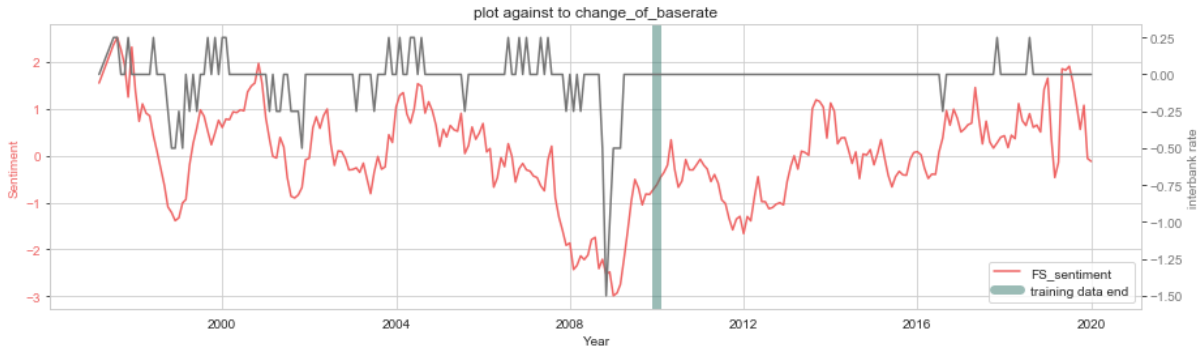

Figure 2.3: FS dictionary sentiment on BOE documents and UK base rate change

The **Figure 2.2** shows the sentiments generated from LM dictionary. And as you may observe, even though the sentiment reflects the 2008 financial crisis as there is a significant drop in sentiment value from 2008 to 2009, LM sentiment in the rest of the periods are merely noise as it failed to explain any rate hike or rate cut in the past 20 years.

The **Figure 2.3** shows the sentiments generated from FS dictionary. And as you may observe, the financial stability dictionary is actually performing well in term of explaining the interest rate movement. We not only observe the significant sentiment drop during the 2008 financial crisis, but also observe the significant sentiment drop in 2012 European Financial Crisis, 2016 Brexit. And all the sentiment drop in 1998, 2002, 2008, 2016 are followed by a series of rate cut in the next few months.

Hence, I observe that financial stability dictionary does have the explanatory power on reflecting the macro-economic performance. So I decided to include some of the words from the Financial Stability dictionary into my self-created BOE interest rate dictionary.

2.2 Region-specific Sentiment Algorithm

To construct a Bank of England interest rate dictionary and derive the sentiment index:

Step 1: Create BOE documents label based on UK base rate

The document download from the website does not have any labels in nature. And normally it would require economic experts to read the documents and judge when this document (minute/speech) is carrying positive and negative sentiment towards the macro-economic situation. However, this is not feasible in my projects and there are 1044 documents for Bank of England alone. And much more are awaiting if includes ECB and FED.

As a result, the document in this thesis is labelled based on local base-rate movement — if there is a base rate rise or drop within the next 6 months after the document published. In this case, if the UK base rate increases within the next 6 month, the document is label as positive sentiment document. If UK base rate drops within the next 6 month, the document is labelled as negative sentiment document. And if neither things happen, the document is then labelled as neutral sentiment document. The rationale is that the base rate has a strong connection with the macro-economical situations. Usually rate cut represents the bank is boosting the economy as the macroeconomic is not good enough as expected. And rate-hike represents the bank is controlling inflation and overheating in the economy, hence macro-economics are performing well.

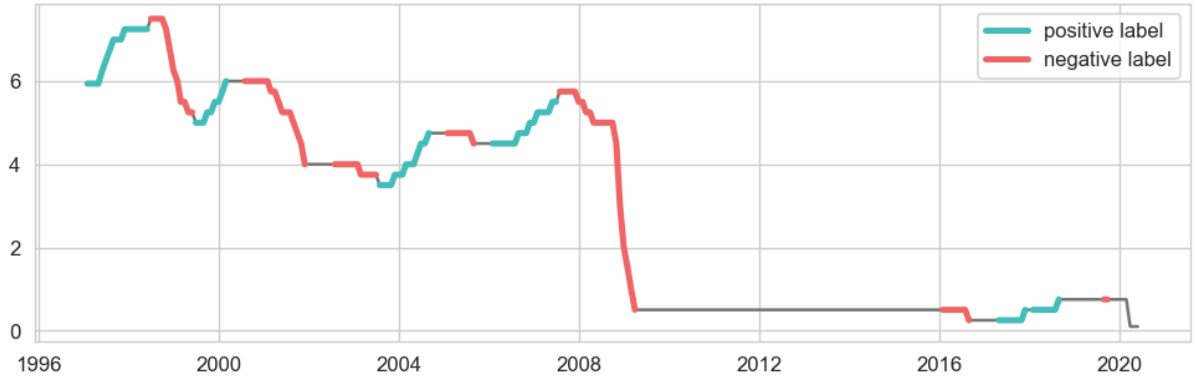

Figure 2.4: Label of the document against the dates of the documents being published

Finally, I split the BOE documents into in-sample and out-of-sample periods. In-sample documents are documents published before 2009-12-31, Out-of-sample (OFS) text are documents published after 2010-01-01. I trained a dictionary based on in-sample and check performance based on OFS documents.

Step 2: Create Co-occurrence matrix on in-sample documents

After the documents being labelled, I calculated the number of times each word occur in a positive/negative/neutral sentiment document that was published before 2009-12-31. Then for each word, for all sentiment-carrying document that contains this word, I calculated the percentage of positive sentiment documents among all documents containing this word. Then the word is labelled as a positive sentiment word if it shows up more frequently in positive sentiment documents than negative sentiment documents. And vice versa for detecting negative sentiment words.

	negative count	positive count	neutral count	p percentage no neutral	n percentage no neutral	sentiment
examin	14	44	9	0.759	0.241	Positive
smooth	15	38	19	0.717	0.283	Positive
exceed	25	52	17	0.675	0.325	Positive
downgrad	11	1	4	0.083	0.917	Negative
turmoil	36	3	2	0.077	0.923	Negative
downsw	4	0	9	0.000	1.000	Negative

Figure 2.5: Word co-occurrence matrix with labelled documents

'negative count' represents the number of negative sentiment documents that contain this word.
'positive count' represents the number of positive sentiment documents that contain this word.
'neutral count' represents number of no-sentiment documents that contain this word.
'p percentage no neutral' represents the percentage of document being positive sentiment among all sentiment-carrying documents that contain this word.

'n percentage no neutral' represents the percentage of document being negative sentiment among all sentiment-carrying documents that contain this word.

Step 3: Select words into dictionary
The BOE dictionary is created based on the parameter space of $(\mathbf{N}, \mathbf{P_{positive}}, \mathbf{P_{negative}}, \mathbf{M_{positive}}, \mathbf{M_{negative}})$:

1. We include **positive sentiment word** into our "BOE dictionary" by the criteria of:
   (a) Showed up in at least **N** pieces of in-sample documents
   (b) 'p percentage no neutral' is greater than $\mathbf{P_{positive}}$
   (c) After sorting on 'p percentage no neutral', we select maximum $\mathbf{M_{positive}}$ words into dictionary
2. We include **negative sentiment word** into our "BOE dictionary" by the criteria of:
   (a) Showed up in at least **N** pieces of in-sample documents
   (b) 'n percentage no neutral' is greater than $\mathbf{P_{negative}}$
   (c) After sorting on 'n percentage no neutral', we select maximum $\mathbf{M_{negative}}$ words into dictionary
3. We include some words from **FS dictionary** into our "BOE dictionary" by the criteria of:
   (a) Positive Sentiment in FS dictionary and positive sentiment in word co-occurrence matrix
   (b) Negative Sentiment in FS dictionary and negative sentiment in word co-occurrence matrix

And I finalised the parameter by optimising the classification accuracy based on in-sample documents:

1. Set-up grids for each of the parameter of $(\mathbf{N}, \mathbf{P_{positive}}, \mathbf{P_{negative}}, \mathbf{M_{positive}}, \mathbf{M_{negative}})$. In this my case, I choose N from [30,50,70], P from [0.55,0.60,0.65] and M from [80,100,120]
2. For each set of parameter, construct the dictionary according to the method above
3. For each document, the document is calculated as a negative document if it contains more negative words than positive words, according to the dictionary produced in the last step, and vice versa.
4. For each dictionary constructed, get the classification results for all in-sample documents, using the method in the last step, and calculate the accuracy rate when comparing to the document label created in step 1.
5. Pick the set of parameters that generates the most accurate classification result. In my case, my final dictionary has the parameter of (50,0.55,0.55,100,100)

Step 4: generate BOE dictionary sentiment index
To generate sentiment for each document:
1. **Positive score:** Sum of 'p percentage no neutral' for all positive words occurred in this document
2. **Negative score:** Sum of 'n percentage no neutral' for all negative words occurred in this document
3. **Document Sentiment:** (Positive score – Negative score) / (Positive score + Negative score)

Step 5: Monthly index by exponentially weighted averaging
The final monthly central bank sentiment index is constructed by first averaging the sentiments of all texts published in the same month (eg: May). Then it is exponentially moving averaged with half-life equalling 3 months. The final sentiment is classified as the sentiment for next month (eg: June)

2.3 Baseline Model Performance

Figure 2.6 shows how does the dictionary looks like. The left includes words that are positive sentiment and right includes words of negative sentiments. The size represents the value of 'p percentage no neutral' or 'n percentage no neutral'. The bigger the word is, the more import this word is in determine the sentiment of documents. We could see that positive word includes: "our-perform" "friendly" "superior", and negative words include: "turmoil" "collapse" "misunderstand" "cutback" "breach". There are noise and classification of the words, but it largely represents the connection of each word to positive and negative sentiment words.

Figure 2.6: Word cloud of BOE interest rate dictionary

Figure 2.7: Newly created dictionary sentiment on BOE documents and UK base rate change

Figure 2.7 shows how does the sentiment index using the **Region-specific Sentiment Algorithm** on raw BOE document looks like[4], plotting against the base rate change from 1997-2020. The green line represents the date where the training periods ends, and the sentiment on the right of green line are generated based on the testing dataset. We could observe that for in-sample periods (left to green line), then sentiment increases drastically when there is going to be a rate hike and drop to -1 when there is going to be a rate drop. However, after the training period end (OFS, right to green line), the fluctuation of the sentiment index becomes smaller and less meaningful. We did not observe a drop in 2016. Hence, we could say implementing the algorithm directly does not work well as the language in the documents is changing. The sentiment important word in 1997-2010 is different from sentiment determining word after 2010. We definitely need some algorithm to clean up the language in the documents and make a dictionary-based model can work well across all time.

[4]I said "raw document" because I use the document directly after pre-processing. In the next section, I will introduce the method to clean up the language of documents and improve the performance

LM Dictionary	Correlation between sentiments calculated and various monthly timeseries					
Size: 229 words	All text	All text OFS	Speech	Speech OFS	Minute	Minute OFS
LIBOR_1Y_change	0.028	-0.069	-0.001	-0.151	0.084	0.174
LIBOR_6M_change	0.044	-0.059	0.009	-0.139	0.084	0.151
LIBOR_3M_change	0.052	-0.027	0.011	-0.103	0.084	0.117
LIBOR_1M_change	0.080	-0.009	0.058	-0.069	0.100	0.106

FS Dictionary						
Size: 250 words	All text	All text OFS	Speech	Speech OFS	Minute	Minute OFS
LIBOR_1Y_change	0.278	-0.010	0.221	-0.110	0.327	0.275
LIBOR_6M_change	0.318	0.018	0.249	-0.098	0.356	0.302
LIBOR_3M_change	0.308	0.009	0.229	-0.107	0.357	0.277
LIBOR_1M_change	0.324	0.007	0.244	-0.102	0.383	0.210

Self-created						
Size: 319 words	All text	All text OFS	Speech	Speech OFS	Minute	Minute OFS
LIBOR_1Y_change	0.351	0.050	0.345	-0.060	0.387	0.344
LIBOR_6M_change	0.403	0.089	0.381	-0.029	0.456	0.403
LIBOR_3M_change	0.402	0.088	0.373	-0.012	0.472	0.376
LIBOR_1M_change	0.392	0.021	0.361	-0.065	0.495	0.274

Figure 2.8: Correlation between sentiment based on different dictionaries with the monthly change ratio of different Sterling LIBOR rate, including LIBOR 1Y, 6M, 3M and 1M rate.

But if we zoom out for now and comparing the sentiment performance across different dictionary, the **Figure 2.8** shows the correlation between sentiments created by different dictionaries and monthly change of LIBOR 1Y 6M 3M and 1M rate. For more information about each column:
All text: represent sentiment from 1997 to 2020, created based on both UK minutes and speech before 2010
All text OFS: represent sentiment from 2010 to 2020, created based on both UK minutes and speech before 2010
Speech: represent sentiment from 1997 to 2020, created based on both UK speech only before 2010
Speech OFS: represent sentiment from 2010 to 2020, created based on both UK speech only before 2010
Minute: represent sentiment from 1997 to 2020, created based on both UK Minute only before 2010
Minute OFS: represent sentiment from 2010 to 2020, created based on both UK Minute only before 2010

Hence for example, in the first table, the value of 0.028 in the top left entry represents the correlation between {sentiment index created by LM dictionary from 1997 to 2020 created based on both minute and speech} and {UK Libor 1Y monthly change from 1997 to 2020} is 0.028. And from these three tables, we could achieve the following conclusions:

1. Context is important for dictionary based model
LM dictionary barely captures any useful sentiments in the central bank statements as its sentiment has almost 0 correlation on all types of LIBOR rate and documents. The FS dictionary, in contrast, has 0.278 correlation with LIBOR 1Y rate based on both minutes and speech. And our BOE dictionary constructed a sentiment with 0.351 correlation with LIBOR 1Y rate, especially on Bank of England Minutes. The difference in performance is expected. LM dictionary is trained on US equity firms annual report, FS is trained on Central Bank financial stability report and our dictionary is trained on BOE interest rate documents directly.

2. Minutes has consistent language styles over time
Both Financial Stability dictionary sentiment and self-created dictionary sentiment based on raw BOE documents have a consistent correlation with Libor 1Y rates. For self-created dictionary based on BOE minutes, it has 0.387 correlation during the full sample periods and 0.344 during the out-of-sample periods with LIBOR 1Y rate. And the same thing happens to the correlation with LIBOR 6M 3M and 1M. We could see the dictionary that was created based on the document until 2009 still have large explanatory power on minutes until 2020. This shows that the vocabulary used in BOE minutes are most of the time similar and unchanged over time.

3. It is harder to capture sentiment on speech
All three dictionaries perform significantly worse after 2009/12/31, and our own model performance completely vanished after the training period. In all three tables, the "Speech OFS" is almost 0 for all dictionaries and all different LIBOR rates. This shows the important word in 1997-2010 is different from the important words from 2010-2020. This could due to the change in language in the speech, whereas the performance of minutes was fairly consistent. This also proves that the language of speech changes significantly over time. The speakers tend to use different vocabularies and different way to speak in different years.

4. Performance slightly varies across different LIBOR rate:
Generically speaking, the model performs the best on LIBOR 6M, which is understandable that the document is labelled based on whether is rate change in the next 6 months. But the correlation to different LIBOR is not significantly different from each other.

Chapter 3

Semantic Language Cleaning

From the previous part, we could judge the consistency of the language by looking at how quick the performance of the dictionary decays in the out-of-sample period. We observe that BOE minutes are consistently well-performed, while the speeches sentiment has a very bad correlation with LIBOR rate during OFS periods. And if you look at the speeches and minutes directly, we could also observe that the language is consistent cover time in minutes while inconsistent overtime in speech.

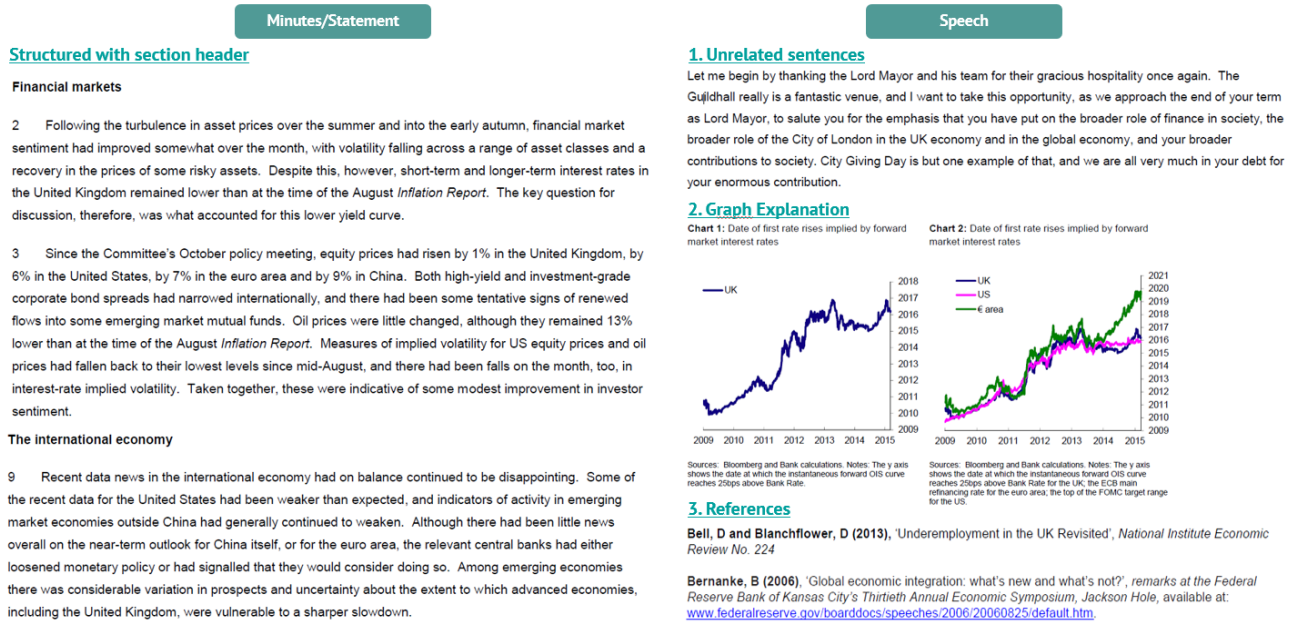

Figure 3.1: Comparison between minutes and speech

As you may observe from the above chart, you could see that the language is standard and consistent for the Bank of England minutes. It has standard header name for each section, which indicates the topics of each paragraph. And the header names are highly similar through the past 23 years from 1997 to 2020. It always talks about inflation, credit, demand/supply and other topics in macroeconomics. But the language in speech is inconsistent and noisy. It contains texts of different format: graph, equations, references, report and unrelated sentences. And the topics changes over time as speech covers more about recent news and less economic contents.

Hence extra actions are needed to clean up the language in documents, to improve the performance of sentiment model. And in this paper, I decide to use Latent Dirichlet Allocation (LDA) topic modelling to remove the less relevant sentences in speech.

3.1 What Is LDA

Latent Dirichlet Allocation (LDA) is a "generative probabilistic model" of a collection of composites made up of parts. Its uses include Natural Language Processing (NLP) and topic modelling, among others. The context of population genetics was proposed by J. K. Pritchard, M. Stephens and P. Donnelly in 2000[1][2]. And LDA was applied in machine learning by David Blei, Andrew Ng and Michael I. Jordan in 2003[3].

Intuitively speaking, LDA is a form of unsupervised machine learning algorithm that find topics in documents automatically, a probabilistic clustering algorithm. Each document can be described by a distribution of topics and each topic can be described by a distribution of words. The LDA topic model assumes:
1. Documents exhibit multiple topics
2. A topic is a distribution over a fixed vocabulary
3. Only the number of topics is specified in advanced
4. All documents are assumed to be generated by **generative process**:
   - Random choose a distribution over topics
   - For each word in the document:
     - Randomly choose a topic from the distribution over topics
     - Randomly choose a word from the corresponding topic (distribution over the vocabulary)

According to the original paper[2], mathematically speaking, if I have a set of D documents, each document having N words, where each word is generated by a single topic from a set of K topics. Hence mathematically, we define d^{th} document is a sequence of N words: $w_d = (w_{d,1}, w_{d,2}, ..., w_{d,N})$. For example, if $w_{d,1} = 3$, this represent in d^{th} document, the word with id=1 show up 3 times. And the **generative process** above can be mathematically written by that, for each document w_d in corpus D, it is generated by for $d = 1 : D$:

1. We define the number of word $N \sim \text{Poisson}(\xi)$, but this is usually prefixed and no need to calibrated
2. We define the variable $\theta \sim \text{Dir}(\alpha)$
3. Then in each word in the document $w_{d,n}$:
   (a) The topic is first chosen by $z_{d,n} \sim$ Multinomial (θ). where θ is a k-dimensional Dirichlet random variable
   (b) Given the chosen topic $z_{d,n}$, we now assign a value to $w_{d,n}$ from a multinomial probability $p(w_n \mid z_{d,n}, \beta)$
4. By completing the loop above, we could now finally generate the document in the vector form $w_d = (w_{d,1}, w_{d,2}, ..., w_{d,N})$

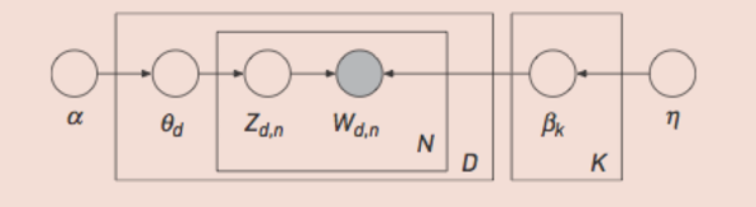

Figure 3.2: Graphical representation of document generation process. The shaded circle is the final document we generated. In LDA, we assume all document are generated by such process.

The **Figure 3.7** visualize the generate process into plot. And mathematically, we can describe the document as following joint distribution:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N} p(z_{d,n} \mid \theta_d)\, p(w_{d,n} \mid \beta_{1:K}, z_{d,n})$$

[1]Pritchard, J. K.; Stephens, M.; Donnelly, P. (June 2000). "Inference of population structure using multilocus genotype data". Genetics. 155 (2): pp. 945–959.
[2]Falush, D.; Stephens, M.; Pritchard, J. K. (2003). "Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies". Genetics. 164 (4): pp. 1567–1587.
[3]Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John (ed.). "Latent Dirichlet Allocation". Journal of Machine Learning Research. 3 (4–5): pp. 993–1022.

$-\beta_{1:K}$ are the topics where each β_k is a distribution over the vocabulary
$-\theta_d$ are the topic proportions for document d
$-\theta_{d,k}$ is the topic proportion for topic k in document d
$-z_d$ are the topic assignments for document d
$-z_{d,n}$ is the topic assignment for word n in document d
$-w_d$ are the observed words for document d, and this is the only observable variable
$-\alpha$ and η are the parameters of the respective dirichlet distributions

Parameter Estimation:
Given the $W_{d,n}$ we could observe for $d = 1 : D \quad n = 1 : N$, we want to estimate the following parameter during the training process:

$-\beta_k$: distribution over vocabulary for topic k
$-\theta_{d,k}$: topic proportion for topic k in document d
$-\alpha$: Distribution related parameter that governs what the distribution of topics is for all the documents in the corpus looks like
$-\eta$: Distribution related parameter that governs what the distribution of words in each topic looks like

One common technique is to estimate the posterior of the word-topic assignments, given the observed words, directly using Gibbs Sampling. More details of Gibbs Sampling for LDA can be found here[4]. It took the original Arthur 15 pages to solve the above equations and mathematically explain how LDA is assigning each document into different topics and assign each topic to different words. As my thesis was not really focused on those theories, but rather about how to implement and use them, I will skip the proof process.

But in term of how to implement it, we can intuitively understand it like that: after the parameter being estimated after training on a set of documents, we could let LDA model describe each topic by a distribution of words. We can use this LDA model to describe the document distribution of topics based on the words it contains. Below shows an example of topic modelling:

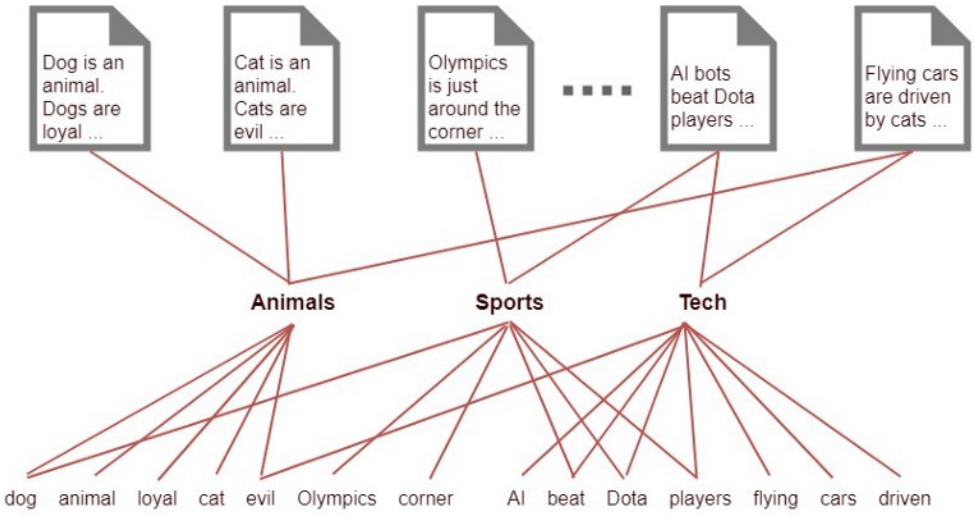

Figure 3.3: Graphical representation of topic modelling process. If we train the LDA model on the 5 documents shown above, the LDA model will learn that the topic animal is linked to dog, animal, loyal, cat, evil. The topic on sports is linked to: dog, Olympics, corner, beat, dota, players. And topic tech is linked to: AI, beat, Dota, Players, evil, flying, cars, driven. Hence, given a new piece of document, the document contains more words that are linked to topic animals, it will higher probability that this document is about the topic on animals than other topics.

[4]Steyvers, M., Griffiths, T. (2007). Probabilistic topic models. In T. K. Landauer, D. S. McNamara, S. Dennis, W. Kintsch (Eds.), Handbook of latent semantic analysis (p. 427–448). Lawrence Erlbaum Associates Publishers.

3.2 LDA For Paragraph Clustering In BOE Minutes

For BOE minutes, even though minutes are structure into different sections, and though for the most time the headers of each section are similar, there are still 166 different unique section headers from 1997 to 2020. And it is tedious to manually cluster all similar headers into one header in order to clean up topics.

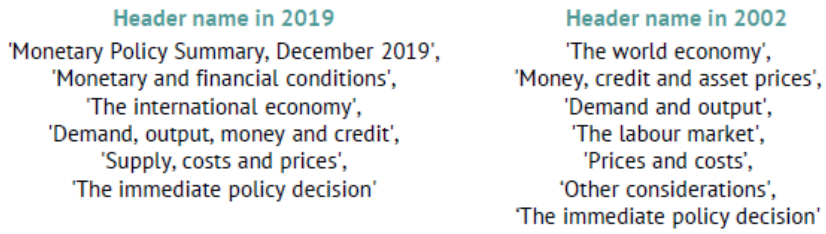

Figure 3.4: Headers are similar but slightly varies

As the picture shows above, I arbitrarily selected two minutes from 1997 to 2020, and it is obvious that those headers are discussing similar topics with just slightly different wording. After I ranked the names of different headers that appeared in last 20 years by frequency, then the 6 most popular headers that appeared in the past 23 years are: "Financial markets", "The immediate policy decision", "Growth and inflation projection", "Money, credit, demand and output", "Supply, costs and prices", "The international economy". Then I aim to rename the headers with other names to one of these headers to clean up the topics.

And following is the algorithms for paragraph cleaning:

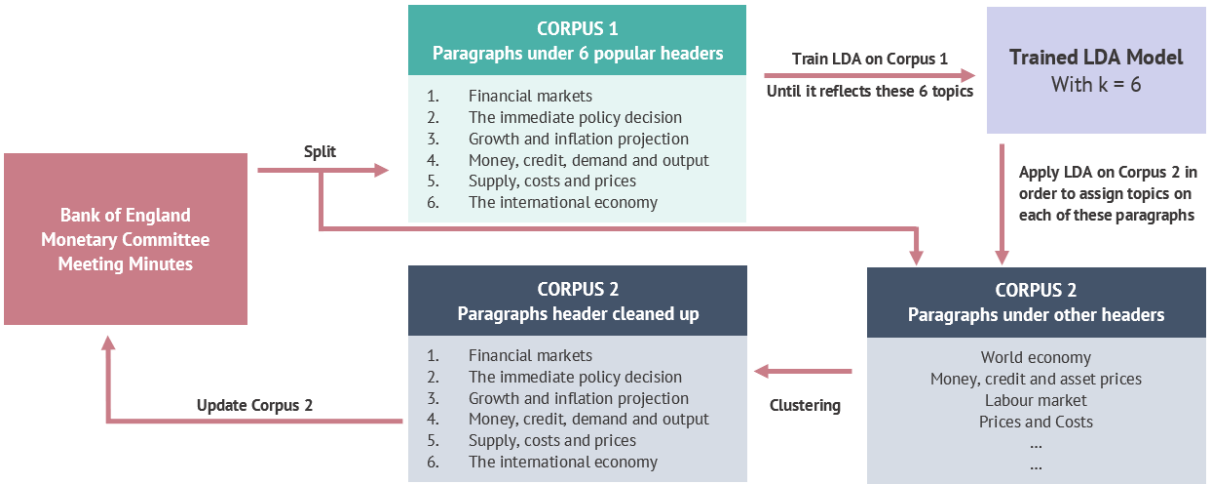

Figure 3.5: algorithm to clean up topics in minutes

1. Setup two set of corpus: Corpus 1 for paragraphs that belong to one of the 6 most popular headers mentioned above. Corpus 2 is for paragraphs that have less popular headers.
2. Train LDA model on corpus 1 with target topics number equals 6. The training result will not be consistent as this is a probabilistic model. Retrain the model again and again until the classification result is overlapped with the 6 most popular headers we already have.
3. Apply the LDA model on corpus 2 to get the classification result for those paragraphs and rename the section header to the classification result.
4. Then all paragraphs in minutes only have section names which belong to one of the 6 topics we mentioned above.

The image below shows the LDA model that has been successfully trained and the word representation for each of the 6 topics that were learnt from the BOE minutes.

Financial markets		The immediate policy decision		Growth and inflation projection	
1. growth	7. data	1. bank	7. issuanc	1. inflat	7. target
2. price	8. survey	2. vote	8. reserv	2. project	8. chang
3. inflat	9. q	3. unanim	9. financ	3. forecast	9. expect
4. remain	10. fall	4. maintain	10. bond	4. rate	10. market
5. labour	11. market	5. purchas	11. stock	5. would	11. period
6. expect	12. indic	6. billion	12. monetari	6. interest	12. mpc

Money, credit, demand and output		Supply, costs and prices		The international economy	
1. output	7. decis	1. monetari	7. stanc	1. concern	7. immedi
2. credit	8. committe	2. appropri	8. incom	2. dollar	8. director
3. discus	9. polici	3. polici	9. job	3. scenario	9. decis
4. money	10. immedi	4. exist	10. mpc	4. oper	10. chief
5. euro	11. market	5. trade	11. circumst	5. reason	11. rapid
6. turn	12. would	6. weaker	12. except	6. market	12. yield

Figure 3.6: The six topics estimated from corpus 1 of BOE minutes

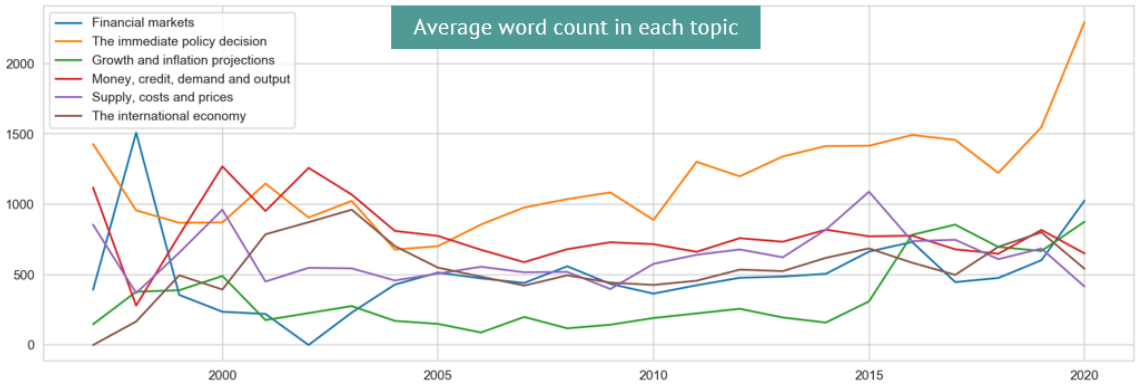

Figure 3.7: Word count of each topic in each minute accross time

And the **Figure 3.7** shows how much focus the BOE minutes are spending on each topic over the past 20+ years. We could first see that the BOE minutes has fairly even distribution on average word-counts of all 6 topics in each minute. This also partially reflect that the language of minutes are consistent in the past 20 years.

In addition, we could observe that the average word count in all topics are gradually increasing in the past 10 years, which means the minutes are getting longer and longer. We could also see that there are significantly growing discussion on the topic 'The immediate policy decision' and 'Growth and inflation projection'.

3.3 LDA For Language Filtering In BOE Speech

As we could see from previous chart **Figure 3.1**, the speech language has lots of non-relevant language and contents. To remove those semantic noise not relevant to Economics, I applied the following procedure:

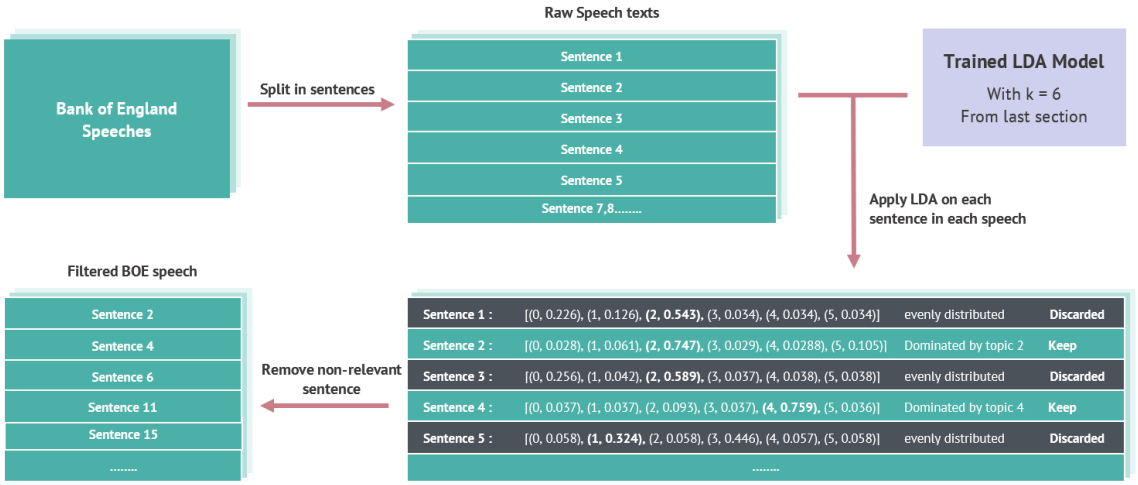

Figure 3.8: Algorithms to remove semantic noise in Speech

Step 1: Raw Noise Removal

We need to first load speech and split each speech into a list of sentences. Speech text is more chaotic than minutes, with many different types of texts inside. The first step is to remove sentences that are obviously not relevant to interest rate, for example: Numbers / mathematics equation / footnotes / references

Step 2: Apply LDA classification on each sentence

I could use the LDA model that is trained on minutes in the previous step and then applying on each sentence of speech. If one sentence has similar language to the minutes that this LDA was trained on, it will fall into one of the 6 topics we have learnt. If the sentence is non-relevant, we could see that we will have an ambiguous classification result.

Step 3: Remove non-relevant sentences and update speech

Here I keep the sentence if the biggest group probability is greater than 0.65. This number was picked up by comparing the sentiment performance[5] from speech with different threshold ranged from 0.5 to 0.8. And I found this threshold gives the best result. And only 36.3% of all words in speech are kept, the rest of the words contain information less useful for interest rate sentiments.

The reason I use the LDA model that was trained on minutes is that I want to remove sentences of speech that have different language style to minutes. As the topics in LDA are trained on minutes and as you can see from **Figure 3.6**, the important words for all these 6 topics are very economical words, such as growth, bank, inflation, monetary and etc. Hence, use this LDA minutes will only a sentences a very confident classification result if this sentence uses the words that are highly overlapping with the vocabularies of one topic in LDA model, which means the language of this sentence is very economical. By confident it means the probability of the sentence is topic k is greater than 0.65, such as [0.9, 0.02, 0.02, 0.02, 0.02, 0.02] which means this sentence has 0.9 probability that this sentiment is under topic 1. And this way of use LDA to identify incoherent language has been proven working by other prior research.[18]

[5]The performance is measured by the correlation between sentiment constructed with the LIBOR 1Y rate. Since different threshold would generate a different corpus for training the model. Hence I change over many different thresholds and pass this corpus to **Region-specific Sentiment Algorithm** to generate a sentiment index specifically linked to this threshold. Then I looked at the in-sample period (from 1997 to end of 2009) sentiment correlation with LIBOR 1Y rate. And finally picked the threshold with the highest correlation

However, the higher the threshold is, the more sentences will be removed from speech, more information will be lost, but the language will be more consistent and similar to the language in minutes. The lower the threshold, the fewer sentences will be removed, the more information is kept, but the language will be less consistent. Hence we need to find a balance between keeping the information and removing irrelevant sentence. This threshold 0.65 is chosen by maximising the training period sentiment correlation with LIBOR 1Y rate after the sentiment constructed based on filtered documents.

And the **Figure 3.9** shows some example of language filtering. We could see sentences mentioned more about inflation or personal opinions are included and less relevant sentences are removed from the dataset.

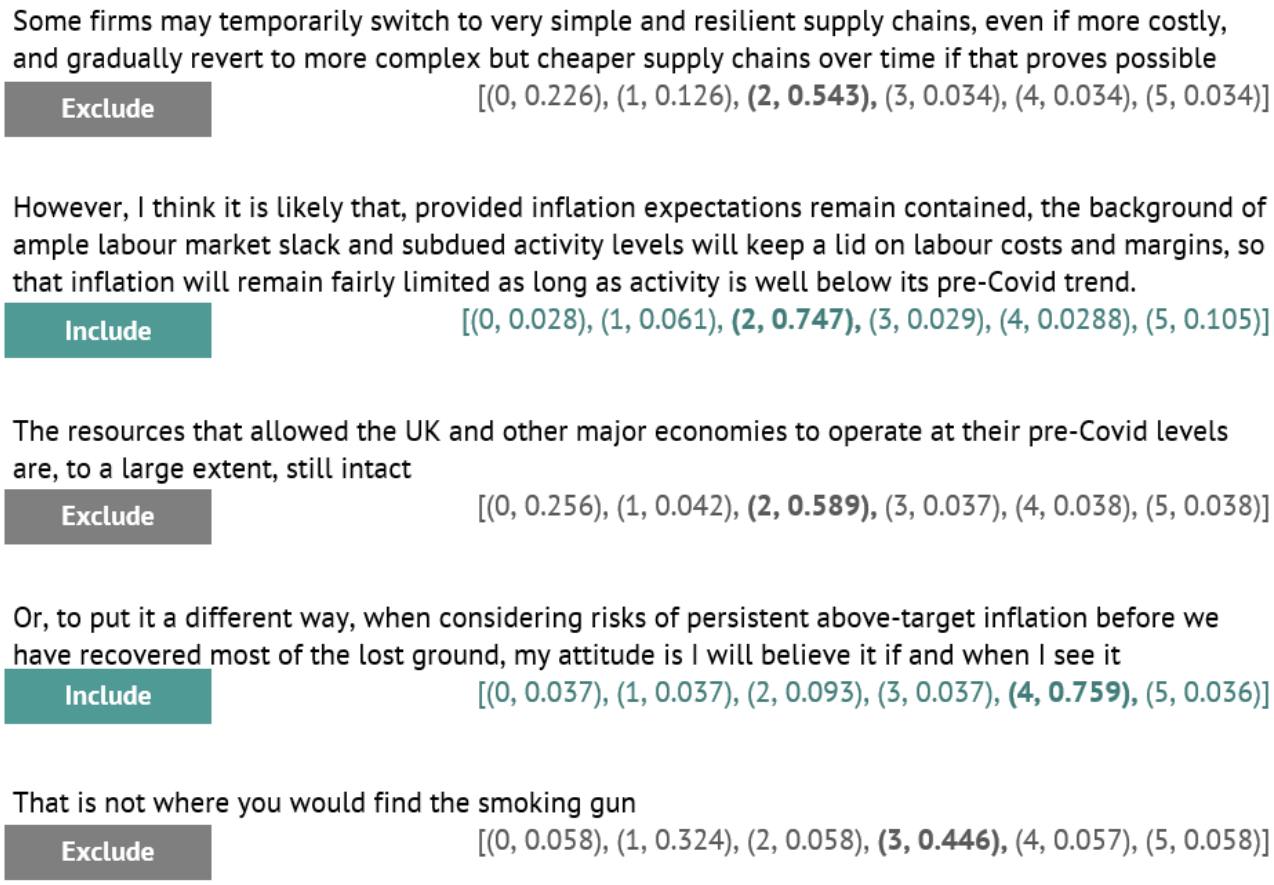

Figure 3.9: Some sentence classification examples

3.4 Language Filtered Model Performance

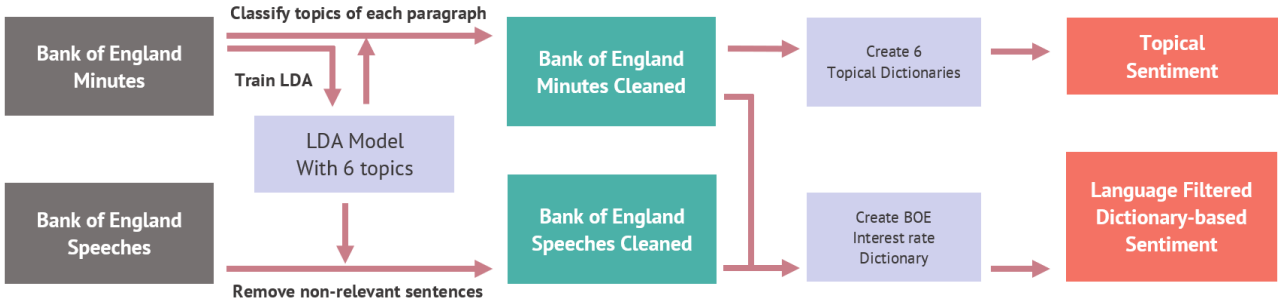

Figure 3.10: Language Filtered Model Summary

The image above shows what we have done so far. In this part, I am going to create two sentiments:
- Interest rate sentiment based on both BOE Minutes and Speech
- Topical sentiment index based on Minutes only by taking advantage of 6 clear topics in each minute

3.4.1 BOE Interest Rate Sentiment

By applying **Region-specific Sentiment Algorithm** from the earlier chapter on the cleaned minutes and speech directly, now we could directly create a monthly dictionary-based interest rate sentiment. To compare the performance across different types of documents, here I created sentiments based on only minutes, only speech and both minutes and speech. The performance is measured on full period (1997-2020) and out-of-sample period (2010-2020). The chart below shows the dictionary created and sentiment performance. The full dictionary can be viewed at the appendix **Figure A.3**.

Figure 3.11: Self-created dictionary word cloud. Left are positive sentiment words, right is negative sentiment

Correlation between all period and OFS period is less fluctuated						
Size: 293 words	All text	All text OFS	Speech	Speech OFS	Minute	Minute OFS
LIBOR_1Y_change	**0.351**	**0.223**	0.346	0.106	**0.367**	**0.341**
LIBOR_6M_change	**0.389**	**0.259**	0.369	0.142	**0.426**	**0.382**
LIBOR_3M_change	**0.360**	**0.218**	0.335	0.122	**0.431**	**0.354**
LIBOR_1M_change	**0.343**	**0.157**	0.325	0.104	**0.450**	**0.275**

Figure 3.12: Dictionary-based method performance on BOE documents after language filtering

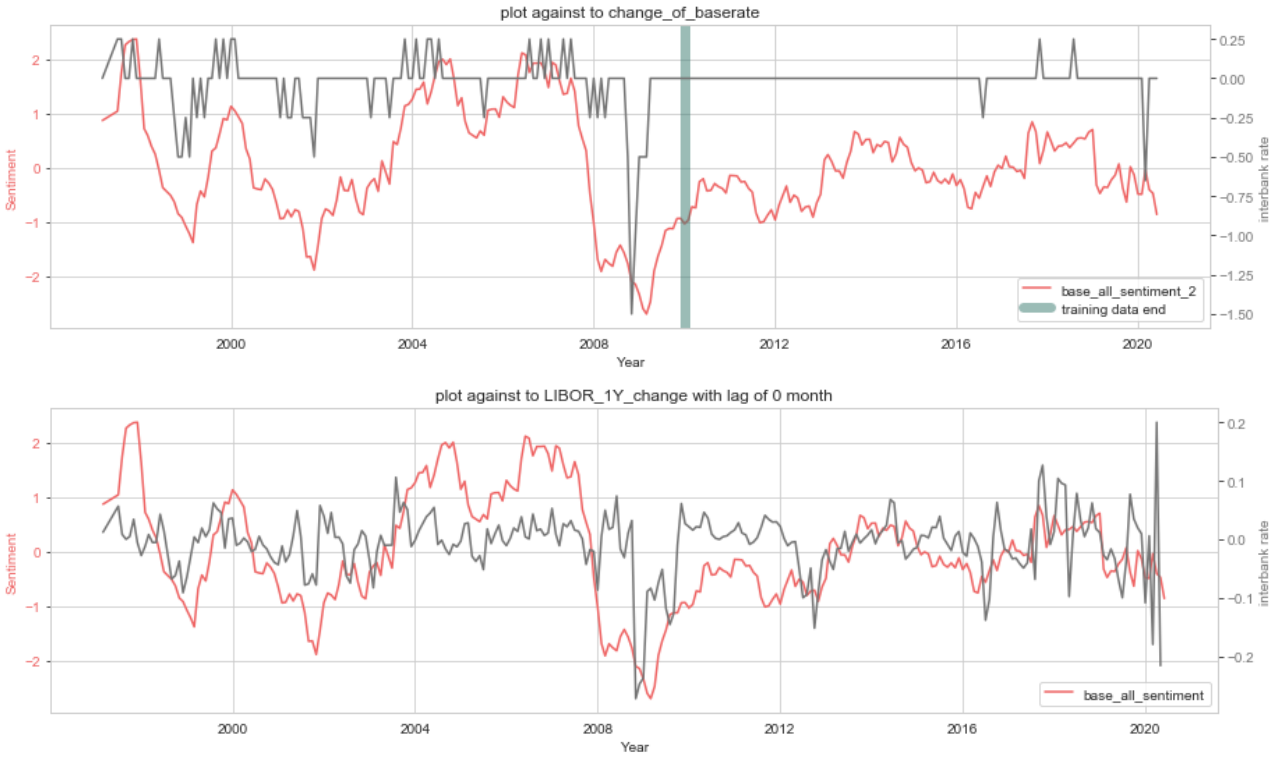

Figure 3.13: Dictionary-based method sentiments with change of base rate (top) and LIBOR 1Y rate (bottom)

From the table above Figure 3.12, which shows the correlation between LIBOR 1Y 6M 3M 1M monthly rate change and the sentiments constructed based on different documents, we could achieve following conclusion:

1. Performance decay less significantly:
The new dictionary, created based on filtered BOE documents, generating a sentiment that shows more consistent correlation during the full sample and out-of-sample periods, comparing to the result from unfiltered documents. The correlation between sentiment and LIBOR 1Y monthly change rate is 0.351 during full sample periods and has 0.223 correlation during the out-of-sample periods. Even though the correlation still drops during the OFS periods, it has been significantly improved comparing to the result based on the un-processed document (see figure 2.8), where the correlation dropped to almost zero during the OFS periods. For example, self-created dictionary sentiment-based raw BOE documents (both minutes and speech) have 0.351 correlation with LIBOR 1Y during the full periods, but only 0.05 correlation during the out-of-sample periods. As a result, we could confidently say that the incoherent language in the speech has been filtered and the sentiment can be constructed without the influence of the irrelevant words. The LDA model make sure the language in speech kept is similar to minutes.

2. Performance slightly varies across different LIBOR rates:
Generically speaking, the sentiment constructed based on different documents are having very similar performance across different LIBOR rate, around 0.36 correlation. But the sentiment is performing the best on LIBOR 6M rate, which is expected, as our document is labelled based on whether the rate is going to change in the next 6 months.

And from Figure 3.13 that plots the interest rate sentiments, base rates and LIBOR rates, we could observe than the sentiment is high whenever that is a increase in Base rate, or the LIBOR 1Y monthly change is positive. And even during the out-of-sample period after 2010, the sentiment movement is still significant. And we observe that the sentiment movement is highly correlated to economical cycle (sentiment drop in 2012 European Financial Crisis, 2016 Brexit and 2020 Covid-19 stock crush.). Hence we could see that the noise in the speech has been removed, because of the more consistent sentiment correlation during the out-of-sample periods and overall sentiment has a better correlation with Libor 1Y rate. In the next few chapters, this algorithm and

this sentiment was further examined about its correlation as well as predicting power to LIBOR 1Y rates as well as to other economic indicators.

3.4.2 BOE Topical Sentiment

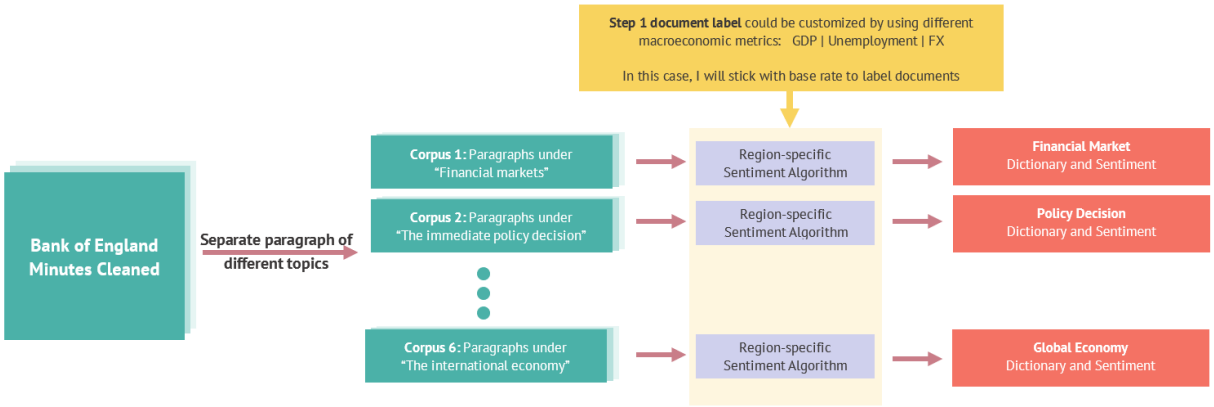

Figure 3.14: Method for creating topical sentiment

Since we have already cleaned the BOE minutes and have set up 6 topics for each document, we can separate paragraphs into 6 different corpora based on their topical allocations. And we can apply **Region-specific Sentiment Algorithm** directly on these 6 corpora to create topical sentiment index and topical dictionary. The **Figure 3.14** shows the workflow of creating topical sentiment from Bank of England minutes. In this case, we have split the content into multiple dimensions – corpus discussing different topics – and it simply requires a suitable model to decode the content into topical information and sentiment.

In this case, I simply use the interest rate to label the document and then create topical sentiment respectively using Region-specific Sentiment Algorithm. The image in the next two pages shows the result of the topical dictionary. From **Figure 3.15**, we could observe that the topical dictionary do reflect the important words in each topics. For example, In "International Economy" dictionary, we do observe German and China represent carries positive sentiment in this dictionary. For "Supply costs and prices", we see 'petrol', 'volatil' and 'depreci' stands for negative sentiment words and 'profit' stands for positive words, which closely related to the commodity market. And from **Figure 3.16**, we could observe that how sentiments of different topics change over time.

Figure 3.15: Vocabularies in topic dictionary after language filtering

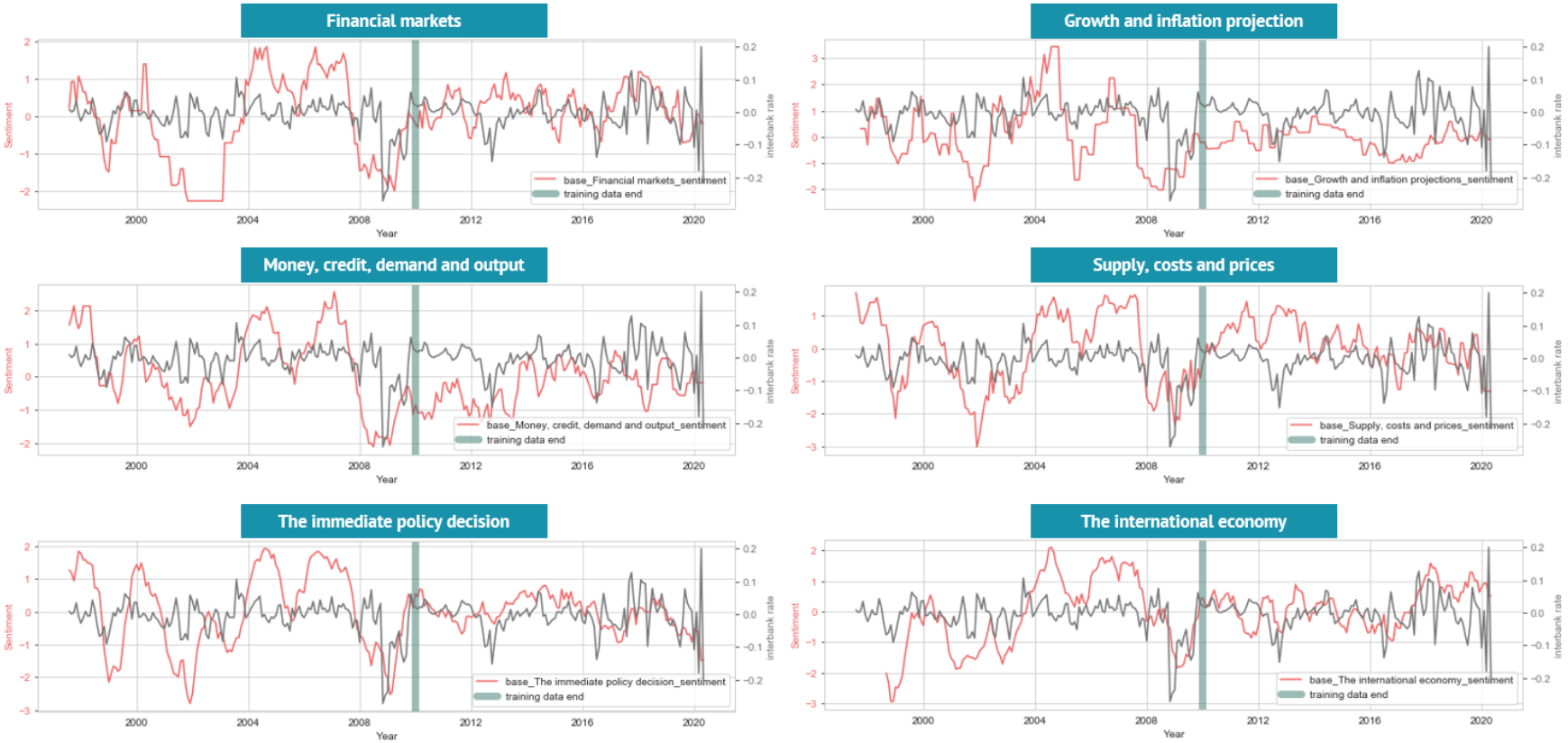

Figure 3.16: Sentiments created by topic dictionary after language filtering

Chapter 4

Model Extensions

In the previous chapter, I have created a BOE interest rate sentiment dictionary (in Section 3.4.1) of 294 words and have also created a sentiment directly using this dictionary. In this chapter, I will show how this dictionary can be extended for other purposes. I will first talk about how to use this dictionary as a dimension-reduction method and to involve machine learning algorithms into the sentiment construction process. Secondly, I will then talk about how this dictionary can be applied on other regions to validate whether the sentiments of the vocabularies have the economical meaning or is this simply a result of over-fitting on Bank of England documents.

4.1 Multinomial Naive Bayesian Model

4.1.1 What Is Multinomial Naive Bayesian Model?

To explain what is Multinomial Naive Bayesian Model, we could start from simple Naive Bayesian model. Naive Bayes is a conditional probability model, where we link the relationship between prior and posterior. According to the Bayes' theorem, given a instance of n features $\mathbf{x} = (x_1, ..., x_n)$, the probability of each of K possible outcomes has the possibility of :

$$p\left(C_k \mid \mathbf{x}\right) = p\left(C_k \mid x_1, ..., x_n\right) = \frac{p\left(C_k\right) p\left(\mathbf{x} \mid C_k\right)}{p(\mathbf{x})}$$

And according to the chain rule for conditional probability:

$$\begin{aligned} p\left(C_k, x_1, \ldots, x_n\right) &= p\left(x_1, \ldots, x_n, C_k\right) \\ &= p\left(x_1 \mid x_2, \ldots, x_n, C_k\right) p\left(x_2, \ldots, x_n, C_k\right) \\ &= p\left(x_1 \mid x_2, \ldots, x_n, C_k\right) p\left(x_2 \mid x_3, \ldots, x_n, C_k\right) p\left(x_3, \ldots, x_n, C_k\right) \\ &= \cdots \\ &= p\left(x_1 \mid x_2, \ldots, x_n, C_k\right) p\left(x_2 \mid x_3, \ldots, x_n, C_k\right) \cdots p\left(x_{n-1} \mid x_n, C_k\right) p\left(x_n \mid C_k\right) p\left(C_k\right) \end{aligned}$$

And starting from this step, the name of 'Naive' come into play: we assume all features in $\mathbf{x}$ are mutually independent. And under this assumption, we have the result that $p\left(x_i \mid x_{i+1}, \ldots, x_n, C_k\right) = p\left(x_i \mid C_k\right)$. And hence the previous equation could be simplified to:

$$\begin{aligned} p\left(C_k \mid x_1, \ldots, x_n\right) &= \frac{p\left(C_k, x_1, \ldots, x_n\right)}{p(\mathbf{x})} \\ &= \frac{p\left(C_k\right) p\left(x_1 \mid C_k\right) p\left(x_2 \mid C_k\right) p\left(x_3 \mid C_k\right) \cdots}{p(\mathbf{x})} \\ &= \frac{p\left(C_k\right)}{p(\mathbf{x})} \prod_{i=1}^{n} p\left(x_i \mid C_k\right) \end{aligned}$$

And in this scenario, k=3 as we have document either being Positive, Negative or Neutral sentiment. $\mathbf{p}(\mathbf{x})$

represent the probability this set of words appeared in the document. And $p(C_k \mid x_1, \ldots, x_n)$ represent the probability the document is positive/negative/neutral sentiment document given that word 1,2 ... n appeared $x_1, \ldots, x_n$ times in this document. The ultimate purposes of multinomial naive Bayesian model is to learn the value of $p(x_i \mid C_k)$ for all words appeared in the document.

4.1.2 Bag-of-word And Why Dimension Reduction Is Needed?

The bag-of-words model is a simplifying representation, such that a text (such as sentence or document) is represented as a list of word-count for each vocabulary that appeared in the documents, disregarding the grammar or word order. For example, suppose we have:

Document 1: Consumer confidence has declined markedly and housing market activity has practically declined.
Document 2: Activity has fallen sharply since the beginning of the year.

We could have the bag of word representation for document 1 of
{'has': 2, 'declined': 2, 'consumer': 1, 'confidence': 1, 'markedly': 1, 'and': 1, 'housing': 1, 'market': 1, 'activity': 1, 'practically': 1}

And for document 2 of
{'the': 2, 'activity': 1, 'has': 1, 'fallen': 1, 'sharply': 1, 'since': 1, 'beginning': 1, 'of': 1, 'year': 1}

Hence, by let column 1 of the Bag of word representation being the first word in the document and column 2 represent the second word in the document, etc, we could have the following bag of word representation in matrix form (2×18)matrix. This means there are 18 different unique vocabularies appeared in 2 documents:

	Activity	Consumer	activity	and	beginning	confidence	declined	fallen	has	housing	markedly	market	of	practically	sharply	since	the	year
0	0	1	1	1	0	1	2	0	2	1	1	1	0	1	0	0	0	0
1	1	0	0	0	1	0	0	1	1	0	0	0	1	0	1	1	2	1

And the same BOG representation can be applied on the whole Bank of England Corpus from 1997 to 2020, then we could have a matrix of 1044 rows, as shown below (**Figure 4.1**)

	rate	inflat	bank	growth	price	polici	market	would	year	economi	...	fodder	foe	fogel	foggi	foggier	reorganist	folder	foley	follo	schiopu
0	14	7	1	5	3	6	16	0	9	7	...	0	0	0	0	0	0	0	0	0	0
1	6	6	31	11	4	8	4	1	8	4	...	0	0	0	0	0	0	0	0	0	0
2	15	13	19	11	13	31	9	14	6	6	...	0	0	0	0	0	0	0	0	0	0
3	42	31	11	24	34	4	16	10	19	3	...	0	0	0	0	0	0	0	0	0	0
4	50	28	3	28	42	7	24	33	23	6	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1039	25	20	80	5	24	29	47	70	15	25	...	0	0	0	0	0	0	0	0	0	0
1040	13	74	14	44	33	34	14	13	11	26	...	0	0	0	0	0	0	0	0	0	0
1041	32	31	78	1	9	37	21	21	3	16	...	0	0	0	0	0	0	0	0	0	0
1042	41	27	61	7	35	37	44	40	14	31	...	0	0	0	0	0	0	0	0	0	0
1043	19	28	2	12	7	14	4	25	17	27	...	0	0	0	0	0	0	0	0	0	0

Figure 4.1: Bag of word representation for BOE documents

The problem with Machine Learning on text data is that the vocabulary size is generally large while the availability of the document is limited. For example, in this case, we have 1044 BOE documents including both speech and minute. However, if we count all unique words in these documents, we could found more than 15000 different words while most of the words have frequency smaller than 10. The **Figure 4.2** shows the word frequency across all BOE documents from 1997 to 2020.

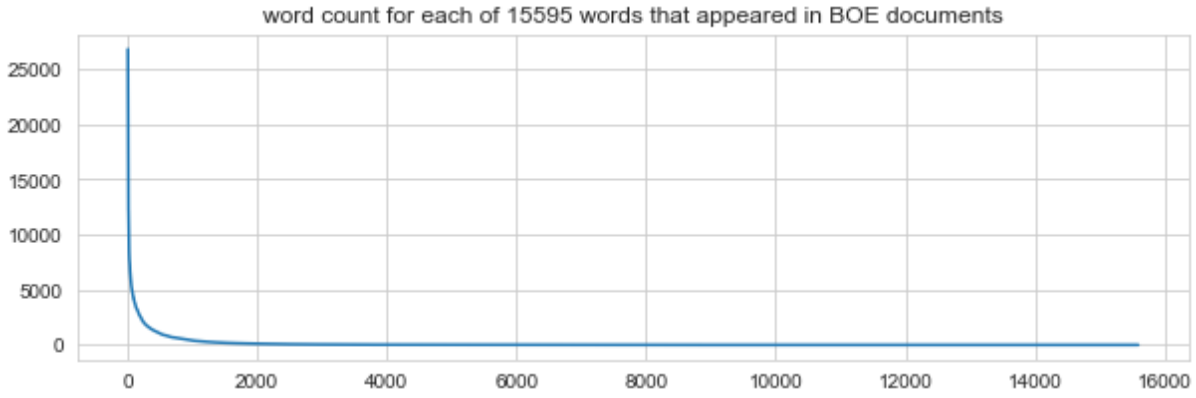

Figure 4.2: Word frequency across all BOE documents

As you may observe, even though there are 15595 words appeared, most of them are very rarely appeared. Hence, most of the vocabularies are irrelevant to the machine learning tasks and the bag of word representation matrix is largely sparse. And we have 15595 different features to be trained on 1044 pieces of documents, which is not enough. Hence a dimension reduction method is needed before passing the Bag-of-word matrix to any machine learning tasks.

4.1.3 TF-IDF Weighting In Document Representation

TFIDF stands for "Term Frequency — Inverse Document Frequency" which could greatly improve the performance of NLP model[1][19] by representation each vocabulary in the document with its importance instead of word-count. This is a technique to compute a weight to each word which signifies the importance of the word in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining[2].

To explain it in more details, I have the following terminology:
- **t** : term (word)
- **d** : document (set of words)
- **N** : count of corpus
- **corpus** : the total document set

Term Frequency (TF):
This measure the frequency of a word in the document. This depends on the length of the documents as well as the usage of the word. For example, the English stop word 'is' or 'the' may appear multiple times in a document. However, in a document of length 100 words and for a document of 10000 words, this is highly likely the number of times the word 'is' appeared in a long document is much much greater than that appeared in a short document. Hence, the term frequency is often divided by the document length as a way of normalization to balance the importance of the word count.

$$tf(t,d) = \frac{\text{count of } \mathbf{t} \text{ in } \mathbf{d}}{\text{number of words in } d}$$

However, simply using TF to represent document is not enough. The main problem is that words which are the most common words such as 'is, are' are still having very high values, but thes words have very low importance. Using these words to compute the relevance produces bad results.

Document Frequency (DF):
This measures the importance of the document in the whole corpus, similar to TF. The only difference is that

[1]You can find the paper at this address. https://www.aclweb.org/anthology/W13-1728
[2]Hiemstra, D. A probabilistic justification for using tfidf term weighting in information retrieval. Int J Digit Libr 3, 131–139 (2000). https://doi.org/10.1007/s007999900025

DF measures the count of occurrences of term t in the document set N, instead of d, which counts how many documents in the whole corpus contain this word.

$$df(t) = \text{number of documents contains } \mathbf{t}$$

Inverse Document Frequency (IDF):
This measures how important a term is. When calculating IDF, the value will be very low for most occurring words such as stop words, because these words occur in almost all documents, df value is large and N/df will give a very low value to that word.

$$idf(t) = log(N/(df + 1))$$

In IDF, for exotic word, the value df would be really small, as only a few documents in the whole corpus is containing this word. Hence, the IDF will be big as N/(df+1) will be big. We use log to control the size of IDF to make sure the value is in range 0 to 8 for the most scenarios.

Term Frequency — Inverse Document Frequency (TF-IDF)
The final TFIDF has the following equation:

$$\text{tf-idf}(t, d) = tf(t, d) * log(N/(df + 1))$$

Hence by applying the above calculation on Bag-of-word matrix, we could have a TFIDF representation of the BOE documents. And in this method, we could see that the TF-IDF representation of each word is the multiplication of IF and IDF. The advantage of this weighting is that it will give more weights to exotic words than Bag-of-word-representation (BOG) could do.

For example, for some exotic words that occurred rarely in the documents but have strong importance, BOG will only represent it as 0 or 1, comparing to the common words such as "is" or "the" will be represented by 30 or 40 depending on the number of occurrence in this document. In TF-IDF representation, because of the IDF component is very small for common words as it occurred in almost all documents, df is large and $log(N/(df + 1))$ is small. Hence $TF \times IDF$ for common words is smaller than the BOG representation. And for exotic words in the corpus, even if tf is small, but since it occurred rarely in the corpus, df is small and $log(N/(df + 1))$ is large. Hence $TF \times IDF$ for exotic words are bigger than the BOG representation. And in this way, we are representing each word by its relative importance rather than simply the frequency. And it would be easier for Machine Learning method to assign weights for each word as the range value of each feature is closer.

4.1.4 Multinomial Naive Bayesian Model Based On BOE Dictionary

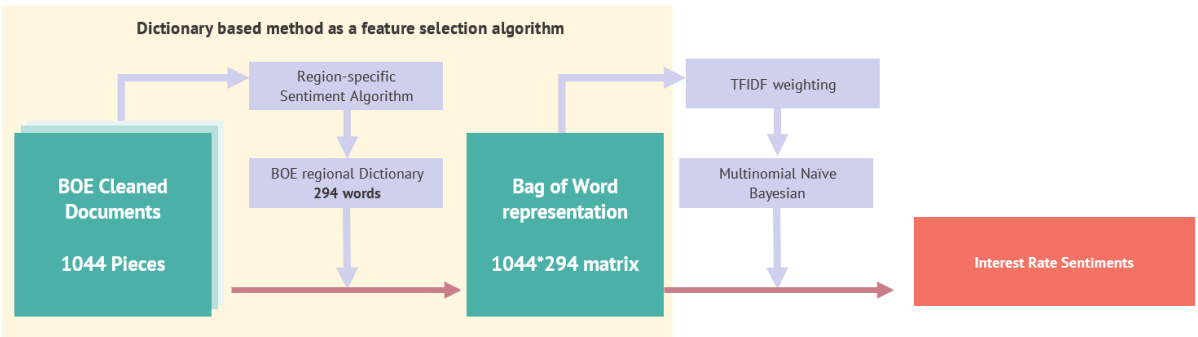

Figure 4.3: Method for creating sentiment using Multinomial Naive Bayesian

The **Figure 4.3** shows the workflow for constructing a Multinomial Naive Bayesian interest rate sentiment based on BOE dictionary that was created based on filtered BOE documents (which was created in Section 3.4.1).

The first step is to use the BOE dictionary to reduce the size of Bag-of-word matrix. As we saw from **Figure 4.1**, the full Bag-of-word matrix has 15000+ columns with only 1044 rows, which is impossible to pass to any machine learning algorithms. By only paying attention to vocabulary that was contained in the BOE dictionary and removing columns that are not contained by the dictionary, we could now have a bag-of-words matrix of size 1044*294, which is then passed to TF-IDF algorithm to get a TFIDF matrix.

And finally, TFIDF representation of the documents is passed to Multinomial Naive Bayesian Model to calculate the interest rate sentiment:

1. I first split the data into training and testing periods. Train the model based on documents from 1997 to 2009 and test the model performance from 2010 to 2020. And I already have the document label for each of the document: 1 -1 0
2. In the Multinomial Naive Bayesian model, each word will learn its importance towards each of the classifications, which is the probability of the document is 1, 0 or -1 given this word occurs.
3. The final prediction is just the label that has the highest probability, given the words that occurred in the document.
4. We could have two types of prediction result
   a. the classification of document 1, -1 or 0
   b. the probability of the document falls into label 1 (positive sentiment)
5. Finally, I applied rolling window on both classification and probabilities results to exponentially weighted average over the sentiments in the past 3 months, in order to get a new sentiment value for the next month.

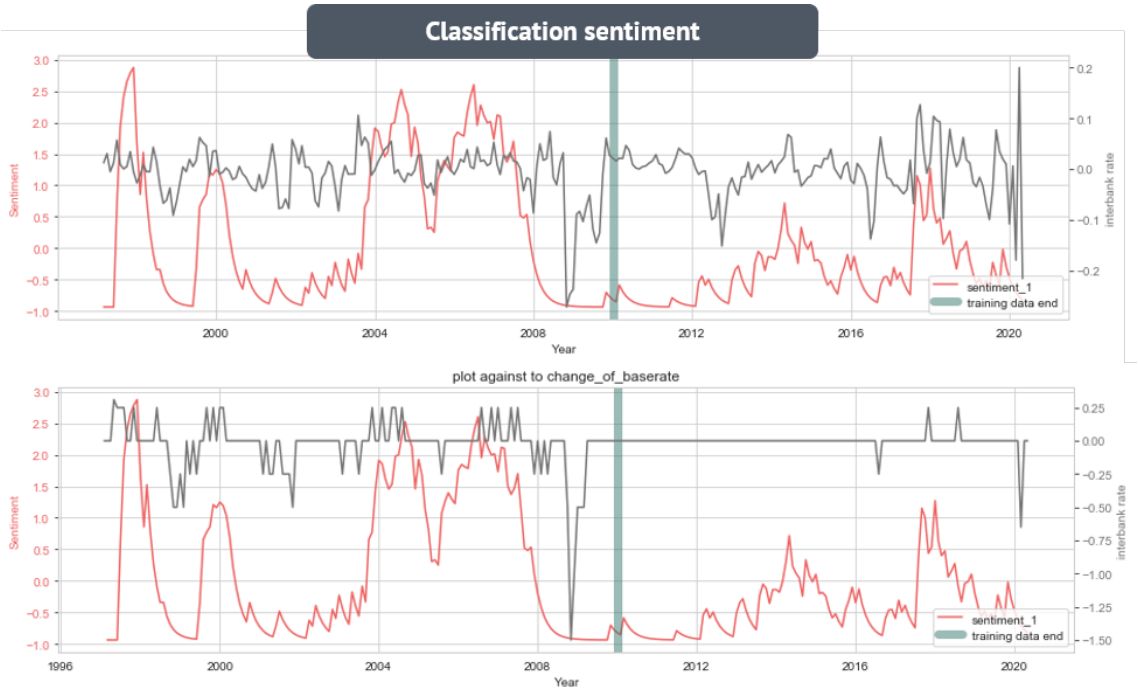

Figure 4.4: Classification sentiment plot against change of UK base rate and LIBOR rate

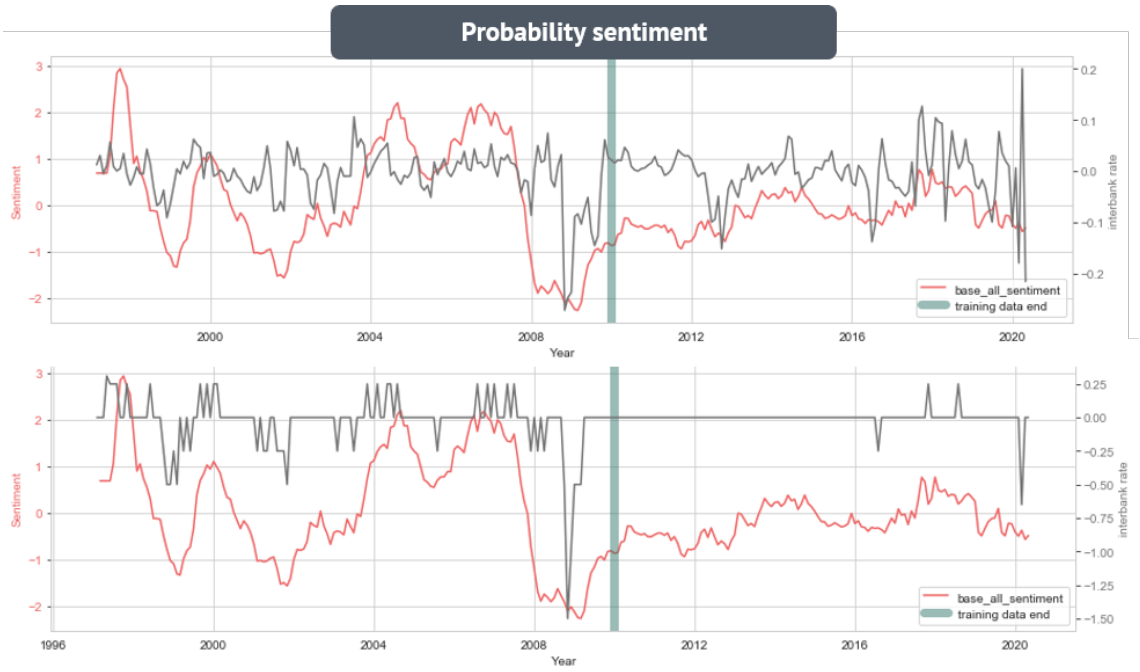

Figure 4.5: Probability sentiment plot against change of UK base rate and LIBOR rate

Classification sentiment

Size: 293 words	All text	All text OFS	Speech	Speech OFS	Minute	Minute OFS
LIBOR_1Y_change	**0.285**	**0.340**	0.271	0.228	**0.297**	**0.324**
LIBOR_6M_change	**0.310**	**0.376**	0.286	0.285	**0.330**	**0.339**
LIBOR_3M_change	**0.303**	**0.379**	0.288	0.367	**0.324**	**0.303**
LIBOR_1M_change	**0.265**	**0.281**	0.261	0.315	**0.301**	**0.179**

Probability sentiment

Size: 293 words	All text	All text OFS	Speech	Speech OFS	Minute	Minute O0FS
LIBOR_1Y_change	**0.344**	**0.260**	0.360	0.039	**0.349**	**0.310**
LIBOR_6M_change	**0.387**	**0.311**	0.388	0.097	**0.403**	**0.347**
LIBOR_3M_change	**0.380**	**0.291**	0.375	0.113	**0.417**	**0.340**
LIBOR_1M_change	**0.362**	**0.186**	0.369	0.080	**0.440**	**0.238**

Figure 4.6: Correlation between various sentiment and different LIBOR rates, including Sterling LIBOR 1Y, 6M, 3M and 1M.

Figure 4.4 shows how the classification sentiment changes over time against monthly change of UK base rate and UK LIBOR 1Y rate. The green line represents the date where the training periods ends, and the sentiment on the right of green line are generated based on the testing dataset. We could observe that for classification sentiment, then sentiment increases drastically when there is going to be a rate hike and drop to -1 when there is going to be a rate drop. We could observe significant sentiment drop in 2008 during the financial crisis, a significant drop in sentiment during the Brexit periods in 2016, and before 2020 corona-virus rate cut. We could observe significant sentiment increase post European Financial crisis in 2014 and before the 2 rate hike by BOE in 2017 and 2018. This shows that the sentiments successfully represents the central bank views on inflation and macroeconomics situation rather than simply the base rate.

And this pattern can also be observed by looking at Figure 4.7, where I plot the classification sentiment from 2010 to 2020 against actual LIBOR 1Y rate. We could observe that the rate is going up when sentiment is up (for example during 2014 and 2018), and going down when sentiment is going down (for example during 2016 and 2019).

Figure 4.4 shows how the probability sentiment changes over time against monthly change of UK base rate and UK LIBOR 1Y rate. The green line represents the date where the training periods ends, and the sentiment on the right of green line are generated based on the testing dataset. We could also see that the sentiment has fairly well performance during the out of sample period to replicate the movement of LIBOR 1Y rate change. This sentiment signal do reflects good macro-economic outlook periods after the 2008 financial crisis, the 2012 European Financial Crisis and after 2016 Brexit periods. And the sentiments turn down whenever there is a crisis coming. Hence this sentiment does perform well in term of interest rate prediction.

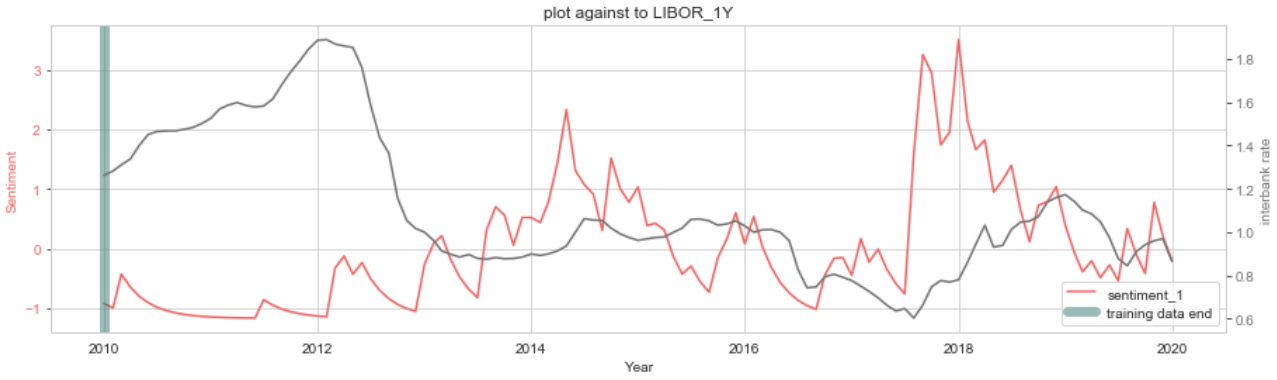

Figure 4.7: Plot of classification sentiment from 2010 to 2020 against UK LIBOR rate

Figure 4.6 shows the correlation between various sentiments and monthly change of LIBOR 1Y 6M 3M and 1M rate. For more information about each column:

All text: represent sentiment from 1997 to 2020, created based on both UK minutes and speech before 2010
All text OFS: represent sentiment from 2010 to 2020, created based on both UK minutes and speech before 2010
Speech: represent sentiment from 1997 to 2020, created based on both UK speech only before 2010
Speech OFS: represent sentiment from 2010 to 2020, created based on both UK speech only before 2010
Minute: represent sentiment from 1997 to 2020, created based on both UK Minute only before 2010
Minute OFS: represent sentiment from 2010 to 2020, created based on both UK Minute only before 2010

Hence, for example, value 0.285 in the top left entry represents the correlation between {sentiment index from 1997 to 2020 created based on both minute and speech} and {UK Libor 1Y monthly change from 1997 to 2020} is 0.285. Value 0.340 on the right represents the correlation between {sentiment index from 2010 to 2020 created based on both minute and speech} and {UK Libor 1Y monthly change from 2010 to 2020} is 0.340.

And from these two tables, we could observe that pattern is the same: the sentiment correlation is consistent for sentiment generated based on minutes and less consistent on sentiment generated based on speech. However, the overall sentiment has around 0.35 correlation during the out of sample periods. This means our sentiment has fairly well performance on reflecting the LIBOR rates movement.

This result shows that the dictionary-based model has the potential to act as a dimension reduction method to limit the number of features in NLP tasks. By using the dictionary, we could only care about the important word in the documents and discard the less relevant word to the interests rate. By adjusting the parameters of $(\mathbf{N}, \mathbf{P_{positive}}, \mathbf{P_{negative}}, \mathbf{M_{positive}}, \mathbf{M_{negative}})$ in **Region-specific Sentiment Algorithm**, we could create a dictionary with adjustable size, and reduce the number of features to a suitable number, instead of being 294 only. And using a Naive Bayesian model is just a simple demonstration of document classification. With the success of this model, we could also try with other ML algorithms such as SVM or random forest, to generate a better interest rate sentiment. But this would be left for future works.

4.2 Cross Region Performance Validation

In the previous part, I have created a BOE interest rate dictionary of 294 words, where each word are stated with a sentiment, 'p percentage no neutral' and 'n percentage no neutral'. In this section, I am going to use this dictionary on other banks' documents to see how the sentiment performed. In the part, I have applied "Step 4: generate BOE dictionary sentiment index" and "Step 5: Monthly index by exponentially weighted averaging" of the **Region-specific Sentiment Algorithm** on 324 FED minutes from 1997 to 2020 and on 285 ECB minutes from 1998 to 2020. And I am comparing the sentiments with their local LIBOR rates, which are US Dollar LIBOR 1Y monthly change and EURO LIBOR 1Y monthly change.

And for each set of documents, we still needs to go through the normal document preprocessing steps:

1. Remvoe punctuation, hyperlinks, references, footnotes, HTML tags, special non-unicode characters, numbers and mathematical equations
2. set all characters into lower cases
3. remove the phrases and convert each word into is base form, in another word, stemming and lemmatization
4. remove all stop words (e.g. for, very, and , of, are, etc) which was chosen from famous NLTK word list.

And here are the results. The advantage of validating the performance of the dictionary using other regions' documents is that we could avoid the over-fitting problem and look forward bias, since I did not use any of these document to train the BOE interest rate dictionary.

Figure 4.8: Plot of ECB sentiment from 2010 to 2020 against EURO LIBOR 1Y rate

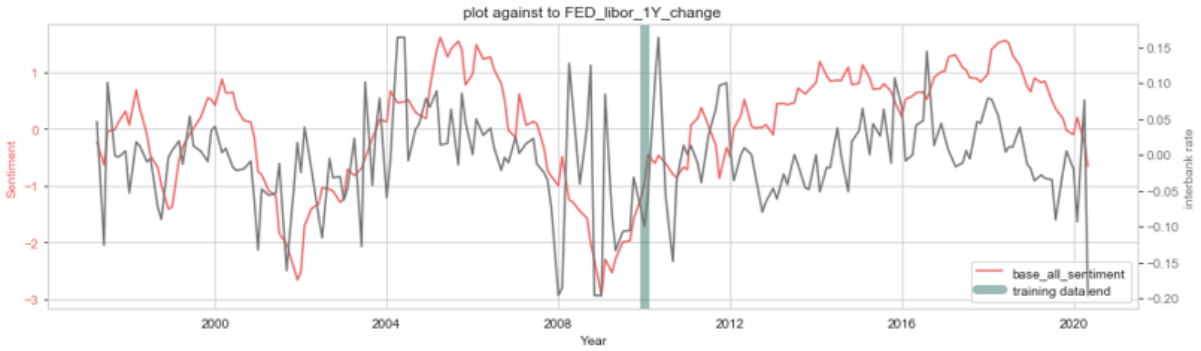

Figure 4.9: Plot of FED sentiment from 2010 to 2020 against US LIBOR 1Y rate

Figure 4.8 shows the sentiment index create based on ECB minutes and press conferences. We could ignore the green line as the whole data-set is out-of-sample data since we did not use these documents to train the dictionary. And we could observe a fairly good amount of correlation visually. We could observe that the sentiment drop significantly in the 2008 financial crisis and the 2012 European crisis. We could also observe the gradual drop of sentiment after the Brexit periods.

Figure 4.9 shows US interest rate sentiment based on FED minutes. We could also observe a significant correlation with LIBOR Rate change visually in the past twenty years. We see a simultaneous drop in sentiment and in LIBOR rate in 2000-2001, 2007-2009, 2019-2020. We could also observe a simultaneous increase in 2002-2006 and 2008-2016.

And if we looking at both figure together, we could further validate the economical power of the BOE dictionary by observing that the European Financial Crisis in 2012 has more impact on ECB sentiment index and limited impact on FED sentiment index. And we could also see a simultaneous sentiment drop in the 2008 financial crisis when this event had more global influence.

ECB data	BOE dictionary	BOE OFS	FS dictionary	FS OFS	LM Dictionary	LM OFS
EUR_LIBOR_1Y	**0.332**	**0.265**	0.311	0.252	0.176	0.235

FED data	BOE dictionary	BOE OFS	FS dictionary	FS OFS	LM Dictionary	LM OFS
US_LIBOR_1Y	**0.368**	**0.174**	0.273	0.074	0.092	0.242

Figure 4.10: Correlation between sentiments created by different dictionary

The Figure 4.10 is comparing the performance of sentiments that were based on different dictionaries. As we could observe, The interest rate dictionary has 0.332 correlation with LIBOR 1Y monthly change rate from 1998 to 2020 and has 0.265 correlation from 2010 to 2020. This interest dictionary also created the US interest rate sentiment that has 0.368 correlation from 1997 to 2020 and 0.174 correlation from 2010-2020. This correlation is much higher than the sentiments generated by FS dictionary and LM dictionary. This result validates that the interested rate dictionary created by this thesis is carrying significant economical meanings and can be applied to other regions as well. And it is more accurate than other existing measures.

Chapter 5

Performance comparing to other existing measures

5.1 Comparison With FinBERT Model

Since this thesis is not focused on deep-learning theory and architecture, I will just briefly talk about the idea behind FinBERT, as this is theoretically complicated with more than 110 million parameters. But I will talk more on how to implement FinBERT for sentence by sentence classification and then to derive a document sentiments on BOE documents in Python. And finally, I will compare FinBERT sentiment with the sentiment created by BOE interest rate dictionary (in Section 3.4.1), to show that my model actually outperforms the complicated algorithms.

5.1.1 What Is FinBERT

FinBERT model is a pre-trained financial sentiment classification model based on "Bidirectional Encoder Representations from Transformers (BERT) model" [20], as you may deduce from its name. BERT makes uses of Transformer [21] architecture, which is an attention mechanism that learns contextual relations between words or sub-words in a text. And Transformer model, in most simple language, consists two sections for language modelling: (1) encoder that reads the text input and (2) a decoder that makes a prediction based on embedded and encoded texts. Contrasting to directional models, which read text/sentences input sequentially (left-to-right or right-to-left), the Transformer encoder layer reads the entire sentence at once, which could be deemed as bidirectional. This allows the model to learn the context of a word based on its surrounding.

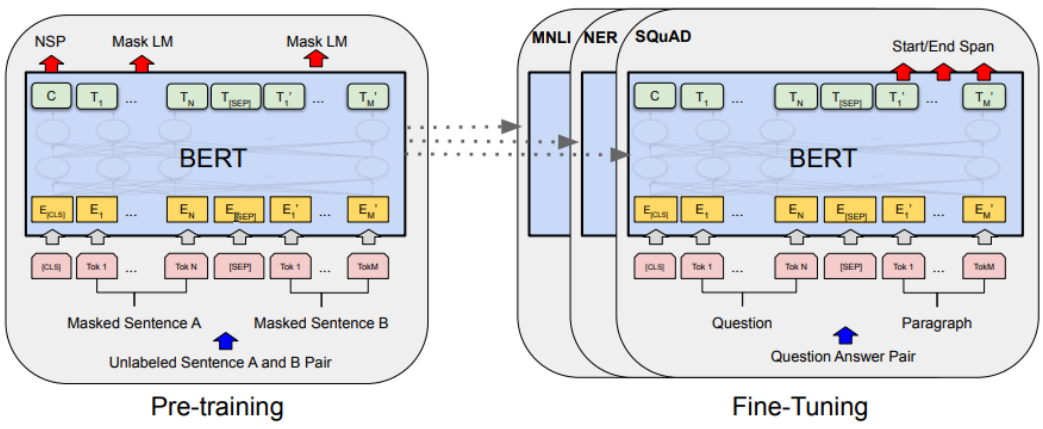

Figure 5.1: Overall pre-training and fine-tuning procedures for BERT.
Source: https://arxiv.org/pdf/1810.04805.pdf

For more details about the theories behind BERT model, please refer to the original paper by **clicking here**. But in terms of how to implement BERT model, it consists of two steps: pre-training and fine-tuning. The pre-training is done on unlabelled documents and Fine-tuning is on labelled text data, as you may see in **Figure 5.1**.

For finBERT model, it follows exactly the same architecture of BERT model but with an additional pre-training step on Reuters TRC2-financial dataset[1], and finally the model conduct the fine-tuning step on Financial PhraseBank dataset [2]. **Figure 5.4** shows how the Phrasebank dataset looks like. It is basically consist of 4845 sentiment-labelled economical and financial sentences. The sentiment is achieved by human judgement with different agreement level. The detailed distribution of agreement level is shown in **Figure 5.3**. And the **Figure 5.2** shows how FinBERT model are pre-trained and fine-tuned.

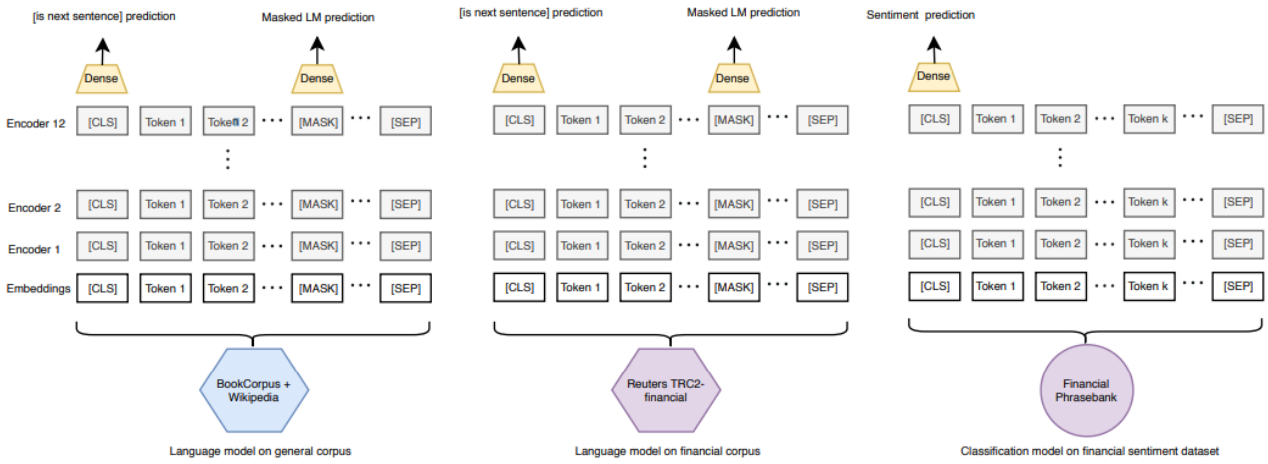

Figure 5.2: FinBERT training Architecture
Source: https://arxiv.org/pdf/1908.10063.pdf

Agreement level	Positive	Negative	Neutral	Count
100%	%25.2	%13.4	%61.4	2262
75% - 99%	%26.6	%9.8	%63.6	1191
66% - 74%	%36.7	%12.3	%50.9	765
50% - 65%	%31.1	%14.4	%54.5	627
All	%28.1	%12.4	%59.4	4845

Figure 5.3: Distribtution of sentiment labels and agreement levels in Financial PhraseBank dataset

5.1.2 How To Calculate FinBERT Sentiment In Python

But in term of implementation of FinBERT model, it is really simple and straightforward.

1. I first downloaded the pre-trained FinBERT model from the following link: Language model trained on TRC2 or Sentiment analysis model trained on Financial PhraseBank. This means the model download from this link has finished the first two steps shown in the **Figure 5.2**.
2. Then I did the fine-tuning of the model on Financial PhraseBank dataset with either 100% agreement level or 75% agreement level. Even though this not same to the language in Bank of England minutes, it is the most similar labelled sentence-by-sentence dataset that I could work on, because I could not find any labelled sentence-by-sentence dataset on central bank communication documents. **Figure 5.4** shows

[1]The TRC2 corpus comprises 1,800,370 news stories covering the period from 2008-01-01 00:00:03 to 2009-02-28 23:54:14 or 2,871,075,221 bytes, and was initially made available to participants of the 2009 blog track at the Text Retrieval Conference (TREC), to supplement the BLOGS08 corpus (that contains results of a large blog crawl carried out at the University of Glasgow). TRC2 is distributed via web download. The data can be requested here https://trec.nist.gov/data/reuters/reuters.html

[2]Data can be downloaded in this link: https://www.researchgate.net/publication/251231364_FinancialPhraseBank-v10

some sample rows of the dataset I am working on. Then the model is ready to be used for sentiment calculation. You can found more details of how to conduct Fine-tuning and prediction on the Finbert official github repo https://github.com/ProsusAI/finBERT. After pretraining and view tuning, you can view FinBERT model a deep learning model with all weights between nodes fixed.

3. To calculate sentiment for each document, I need to first split one document into sentences. For example, the minutes 'Bank Rate maintained at 01% - May 2020' can be split into 327 sentences. And I will perform sentiment calculation on each sentence.
4. On each sentence, the model will calculate the probability of this sentence being positive-sentiment, negative-sentiment, or neutral-sentiment sentence, as shown in the 'logit' column in **Figure 5.5**. The sentiment for each sentence is then: Sentiment = $\mathbf{P}$(positive) − $\mathbf{P}$(negative)
5. The document sentiment is just the average sentiment of all 327 sentences in this minutes.
6. The monthly sentiment index constructed in the same method as before: calculate monthly averaged sentiments and then exponentially weighted averaged with the sentiments of 3 months.

sentence	sentiment
The purchase price will be paid in cash upon the closure of the transaction , scheduled for April 1 , 2009	neutral
With this , the company will exit the contract manufacturing service segment	neutral
Commission income fell to EUR 4.6 mn from EUR 5.1 mn in the corresponding period in 2007	negative
Finnish media group Talentum has issued a profit warning	negative
The loss for the third quarter of 2007 was EUR 0.3 mn smaller than the loss of the second quarter of 2007	positive
Consolidated operating profit excluding one-off items was EUR 30.6 mn , up from EUR 29.6 mn a year earlier	positive

Figure 5.4: Example of the fine-tuning dataset

	sentence	logit	prediction	sentiment_score
0	Authorities around the world are taking action to halt the spread of the Coronavirus (Covid-) pandemic and to support economic activity.	[0.3477197, 0.0023594745, 0.6499208]	neutral	0.345360
1	The Bank of England s Monetary Policy Committee (MPC) sets monetary policy to meet the % inflation target, and in a way that helps to sustain growth and employment.	[0.01741305, 0.0008709934, 0.981716]	neutral	0.016542
2	In that context, its challenge is to respond to the severe economic and financial disruption caused by the spread of Covid-.	[0.003241552, 0.0022447987, 0.99451363]	neutral	0.000997
3	At its meeting ending on May , the MPC voted unanimously to maintain Bank Rate at .%.	[0.35198066, 0.0044658915, 0.64355344]	neutral	0.347515
4	The Committee voted by a majority of - for the Bank of England to continue with the programme of billion of UK government bond and sterling non-financial investment-grade corporate bond purchases, financed by the issuance of central bank reserves, to take the total stock of these purchases to billion.	[0.3056326, 0.0016280895, 0.6927393]	neutral	0.304004

Figure 5.5: Example of sentence by sentence prediction on BOE documents

5.1.3 Why Choose FinBERT As Benchmark

FinBERT[1] model is the most recent (Aug 2019) and most powerful pre-trained language model for NLP tasks on financial documents. It outperform many existing advanced language model on classification task of 'Financial PhraseBank dataset'[22]. The comparing algorithms including:

1. **LSTM classifier:** Use a LSTM model with a hidden size of 128 is used and with the last hidden state size being 256 due to bidirectionality. A dropout probability of 0.3 and a learning rate of 3e-5 is used. The word in the model was using GloVE embeddings.[23] [3]

[3]GloVE Word embedding model can be downloaded directly here: https://nlp.stanford.edu/projects/glove/

2. **LSTM with ELMo:** Use exactly the same LSTM model architecture as above. The only difference is that this model is using ELMo embedding instead.[4]
3. **ULMFit:** Universal Language Model Fine-tuning for Text Classification [24]
4. **LPS:** Model proposed by Pekka, Ankur 2014. [22]
5. **HSC:** Model proposed by Srikumar Krishnamoorthy. 2018 [25]
6. **LPS:** Model proposed by Macedo, Freitas and Siegfried 2018. [26]

Figure 5.6 shows that FinBERT model outperform all 6 other methods significantly on economical content of Financial PhraseBank dataset with 86% accuracy on full dataset and 97% accuracy on 100% agreement data [1]. Hence, using FinBERT as a benchmark to validate model performance will be a great choice as this is one of the most advanced model in the industry. And if my model can outperform finBERT, my model could then be proved to adding new value to Economical sentiment extraction territory.

	All data			Data with 100% agreement		
Model	Loss	Accuracy	F1 Score	Loss	Accuracy	F1 Score
LSTM	0.81	0.71	0.64	0.57	0.81	0.74
LSTM with ELMo	0.72	0.75	0.7	0.50	0.84	0.77
ULMFit	0.41	0.83	0.79	0.20	0.93	0.91
LPS	-	0.71	0.71	-	0.79	0.80
HSC	-	0.71	0.76	-	0.83	0.86
FinSSLX	-	-	-	-	0.91	0.88
FinBERT	**0.37**	**0.86**	**0.84**	**0.13**	**0.97**	**0.95**

Figure 5.6: Classification Results on the Financial PhraseBank dataset.
Source: https://arxiv.org/pdf/1908.10063.pdf

5.1.4 FinBERT Model Performance

I implemented the algorithm showed in Section 5.1.2 on BOE language-filtered minutes and speeches that was consturcted in Section 3.2 and Section 3.3, we could construct a FinBERT monthly interest rate sentiment.

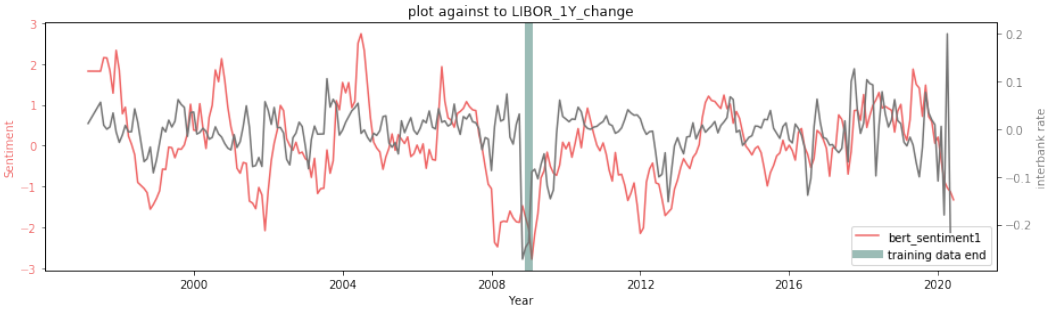

Figure 5.7: Plot of FinBERT sentiment against LIBOR 1Y rate

[4]Full code can be found here: https://colab.research.google.com/drive/13f6dKakC-0yO6_DxqSqo0Kl41KMHT8A1

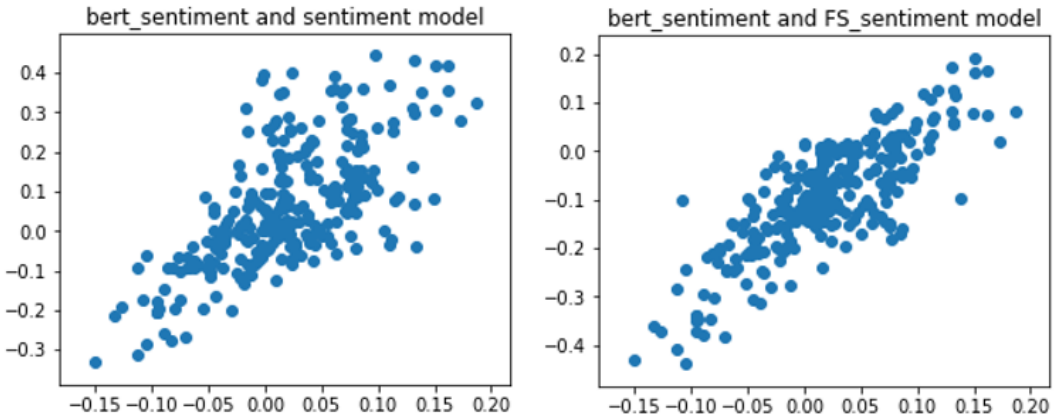

Figure 5.8: scatter plot of FinBERT sentiment with my self-created dictionary sentiment and FS dictionary sentiment. Left is self-created sentiment and right is FS dictionary sentiment

Figure 5.7 shows how the FinBERT sentiment change over time against LIBOR 1Y rate. We could also ignore the green line as the FinBERT model was not trained on any of the BOE documents. Hence there is no train/test data split in this sentiment index. And looking at the sentiment it created, we could observe the significant sentiment drop during the financial crisis in 2008 and 2012. And we could also observe the drop in sentiment whenever there is a significant LIBOR rate drop in 1999, 2002, 2009, 2012, 2016 and 2020. Hence from the sentiment index itself, we could visually prove that the sentiment created by FinBERT model are reflecting the macro-economic cycle and interest rate movement across the past 20 years.

Figure 5.8 shows how similar are the sentiments created by FinBERT model, self-created BOE dictionary and FS dictionary. And as you may observe in the plot where I plot the scatter graph of FinBERT-BOE and FinBERT-FS, the FinBERT sentiment is actually more similar to FS dictionary sentiment, but still highly similar to BOE dictionary sentiment as all the point lines up in a straight line. But this on the other hand also shows that the implementation of this complicated pre-trained model does not make a difference to the sentiment it constructed.

	FinBERT	FinBERT OFS	BOE Dict	BOE Dict OFS	LM Dict	LM Dict OFS	FS Dict	FS Dict OFS
LIBOR_1Y_change	0.283	0.225	0.336	0.239	0.040	-0.066	0.255	0.000
LIBOR_6M_change	0.335	0.264	0.374	0.279	0.060	-0.056	0.288	0.030
LIBOR_3M_change	0.326	0.253	0.350	0.248	0.072	-0.028	0.263	0.021
LIBOR_1M_change	0.346	0.280	0.339	0.203	0.095	-0.016	0.271	0.022

Figure 5.9: Correlation with various UK LIBOR rate

Figure 5.9 shows the correlation between various sentiment and LIBOR rates. We could see that for FinBERT model have a fairly well level of correlation with LIBOR rate across in both full sample periods and out of sample periods. Take an example of LIBOR 1Y rate, the FinBERT sentiment has a correlation of 0.283 from 1997-2020 and a correlation of 0.225 from 2010-2020 (OFS), the result is much better than FS dictionary sentiment where the correlation is almost zeros during the OFS periods. But the correlation is still lower than the BOE dictionary where we achieve 0.336 correlation during the full sample periods. Hence, I could say even if FinBERT model is extracting a lot of information from central bank documents, the performance is not as good as the BOE interest rate sentiment proposed in this thesis. My BOE dictionary sentiment outperforms FinBERT by 18% and outperform FS dictionary sentiment by 32%. Again, complicated model not necessarily performs better, because of lack of available labelled economical sentence-by-sentence dataset.

5.2 Forecasting With Sentiment Indexes

In this section, I will investigate whether a model with text sentiment included will outperform a similar model without the usage of text data. I will compare the prediction power of various sentiment indexes and existing financial indicators x including:

1. Dictionary-based sentiment using Loughran and McDonald (2011) dictionary
2. Dictionary-based sentiment using financial stability dictionary
3. Dictionary-based sentiment using self-created BOE interest rate dictionary 3.4.1
4. Dictionary-based topical sentiment created in BOE topical dictionary 3.4.2
5. Sentiment created by pre-trained deep-learning-based FinBERT Model [1]
6. Sentiment created by python NLP library 'Textbolb'[5]
7. Economic Policy Uncertainty Index[27] [6]

I will evaluate the forecasting power of these text metrics using the following multivariate linear model:

$$y_{t+h} = \alpha + \beta \cdot y_{t-1} + \eta \cdot x_{t-1} + \epsilon_t$$

Where y represent the monthly time-series we want to forecast, t represents each timestamp of the time-series. h represent the forecasting horizon. If $h = 0$, this represents we are forecasting 1 month ahead using the past targeted time-series and the various past information x. To compare model performance without the text information, we simply need to set $\eta = 0$ (the null model, AR(1) model). Then the final predicting power of each x is quantified by comparing the out-of-sample forecasting RMSE(root mean square errors) in a text-data-included model, with that in the text-data-excluded model. If RMSE is much smaller in the model that includes the text information x, we could say this x is helpful in predicting the futures. The text information is most powerful if it has the most significant decrease in RMSE comparing to the null model. In another word, the optimal text sentiment should minimize:

$$\frac{\text{RMSE}}{\text{RMSE}_{\text{NULL}}}$$

And I also looked at different horizons to predict current month, 1, 3, 6 and 9 months ahead ($h = 0, 1, 3, 6, 9$) to get a range of accuracy as well as to construct a confident interval of RMSE ratio. The economical metrics that I am predicting on (y) includes:

1. LIBOR_1Y: UK LIBOR 1Y rate, monthly frequency
2. VIX Close: Monthly close price of CBOE Volatility Index
3. UK_CPI: Monthly Consumer price inflation in the UK
4. UNRATE: UK unemployment rate, quarterly frequency
5. SP500_PE: The historical SP 500 P/E Ratio, monthly frequency
6. BCI: The Business Confidence Index (BCI) in the UK, monthly frequency
7. CCI: The Consumer Confidence Index (BCI) in the UK, monthly frequency
8. GDP: UK Gross domestic product, quarterly frequency
9. GDP_P: UK Gross domestic product per capita, quarterly frequency

[5] To calculate Textbolb sentiment, I first split the document into sentences. And on each sentence, I import this python package directly and use the build-in method to calculate sentiment of each sentence directly. Then the sentiment of the document is just the average of sentiments of all sentences. Finally, the monthly sentiment for the next month is calculated by first averaging sentiments of all documents published in this month, and then exponentially weighted average with the monthly sentiments in the previous 3 months. You may find more information about textbolb from this official page: https://textblob.readthedocs.io/en/dev/

[6] This is completely a separate research done by Scott R. Baker, Nicholas Bloom and Steven J. Davis. They quantify the economical uncertainty based on newspaper coverage frequency. More specifically, this index calculated the number of times when the articles in 10 major US newspaper contain following terms: "economic" or "economy"; "uncertain" or "uncertainty"; "Congress", "deficit", "Federal Reserve", "legislation", "regulation", or "White House". And Index is instantly available and can be downloaded from source: https://www.policyuncertainty.com/

The Figure 5.11 shows the forecast performance of the text model. We could observe that both the overall sentiment and topical sentiment are offering significant forecast improvement for LIBOR 1Y rate, comparing to the null AR(1) model that excludes these sentiments (by looking at the plot at the bottom left in Figure 5.11). We could observe that the model with BOE dictionary sentiment, when horizon equals 0 (use the data last month predict data this month), only has 37.4% of the original RMSE of the Null model without the BOE sentiment. This again proves that the BOE dictionary successfully reflects Central Bank's view on current economics or interest rate level, and this sentiment successfully captures the LIBOR movement. And if we compare the forecasting power of different sentiments (on LIBOR_1Y), we could see that BOE dictionary outperforms FinBERT sentiment again as it has a lower RMSE ratio, where the later has 44.8% RMSE comparing to the Null model. And this statistics is 52.9% RMSE ratio when using FS dictionary sentiment, which is even worse. We could see the BOE interest rate dictionary sentiment do significantly outperforming other existing methods. And in term of predicting the LIBOR 1Y rate, the topical sentiment, "The immediate policy decision", are actually have a better predicting power on LIBOR rate movement with only 33.5% RMSE comparing to the Null model. Hence, I would recommend using this topical sentiment on LIBOR rate forecast tasks.

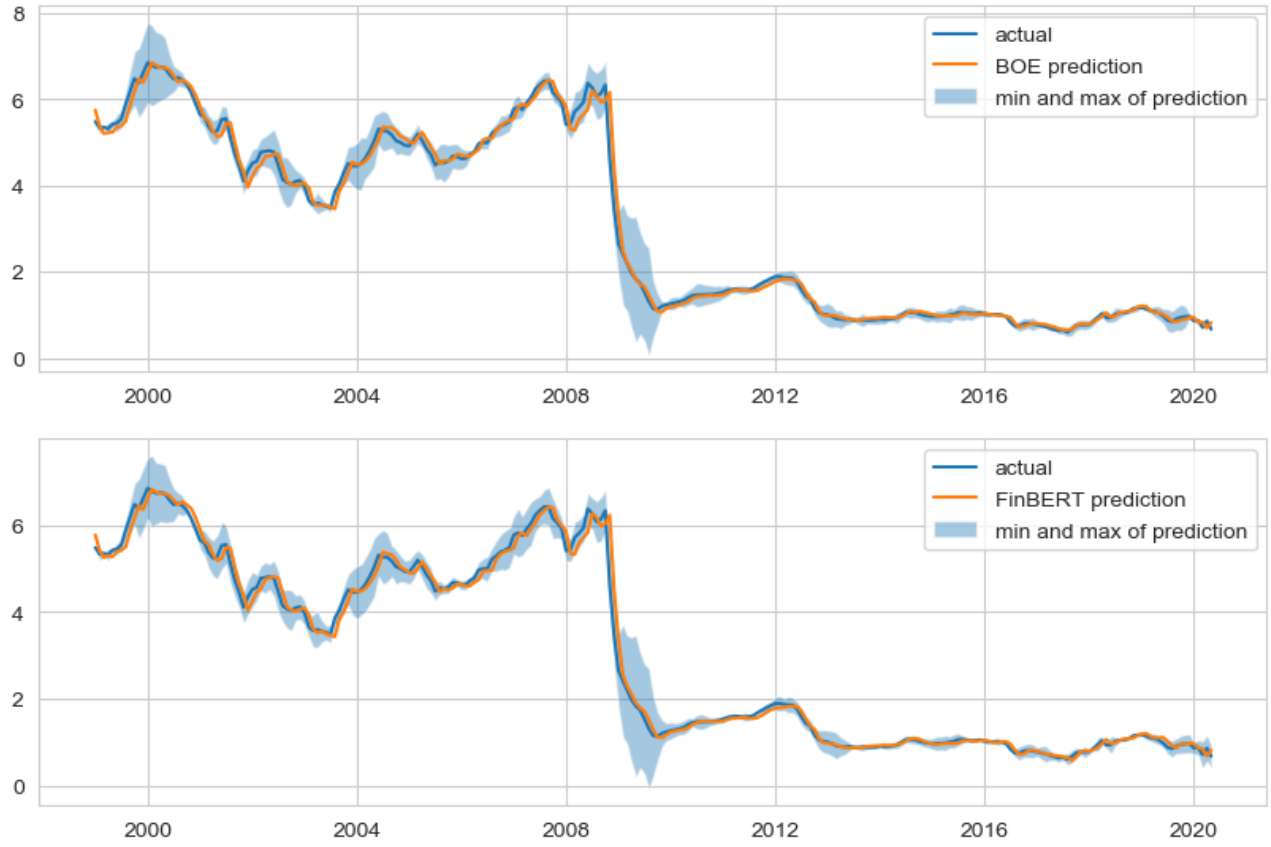

Figure 5.10: Result from the forecasting regression model mentioned in this section. The plot shows the actual LIBOR 1Y rate and predicted LIBOR rate using the regression model mentioned above using (1) BOE interest rate dictionary sentiment (2) FinBERT sentiment as x, both with horizon =0. The confidence interval represent the min and max of predicting value over different horizon (0,1,3,6,9 months ahead)

And the Figure 5.10 compares the forecasting and confidence interval of predicting model based on BOE interest rate sentiment and FinBERT sentiment. From these two plots, we could observe that even the prediction are similar when the horizon is to predicting next month, the FinBERT sentiment model has a larger min/max range if we are predicting on a different horizon. The shaded area is visually larger than that of BOE sentiment across all time, if looking at 2006-2009, 2003-2004. Hence, this again shows that the algorithm proposed by this thesis is performing better than the FinBERT model in term of understanding sentiments carried by the central bank documents. And the complicated model and its 110 million parameters are not making a difference to

this task because of lack of labelled training data.

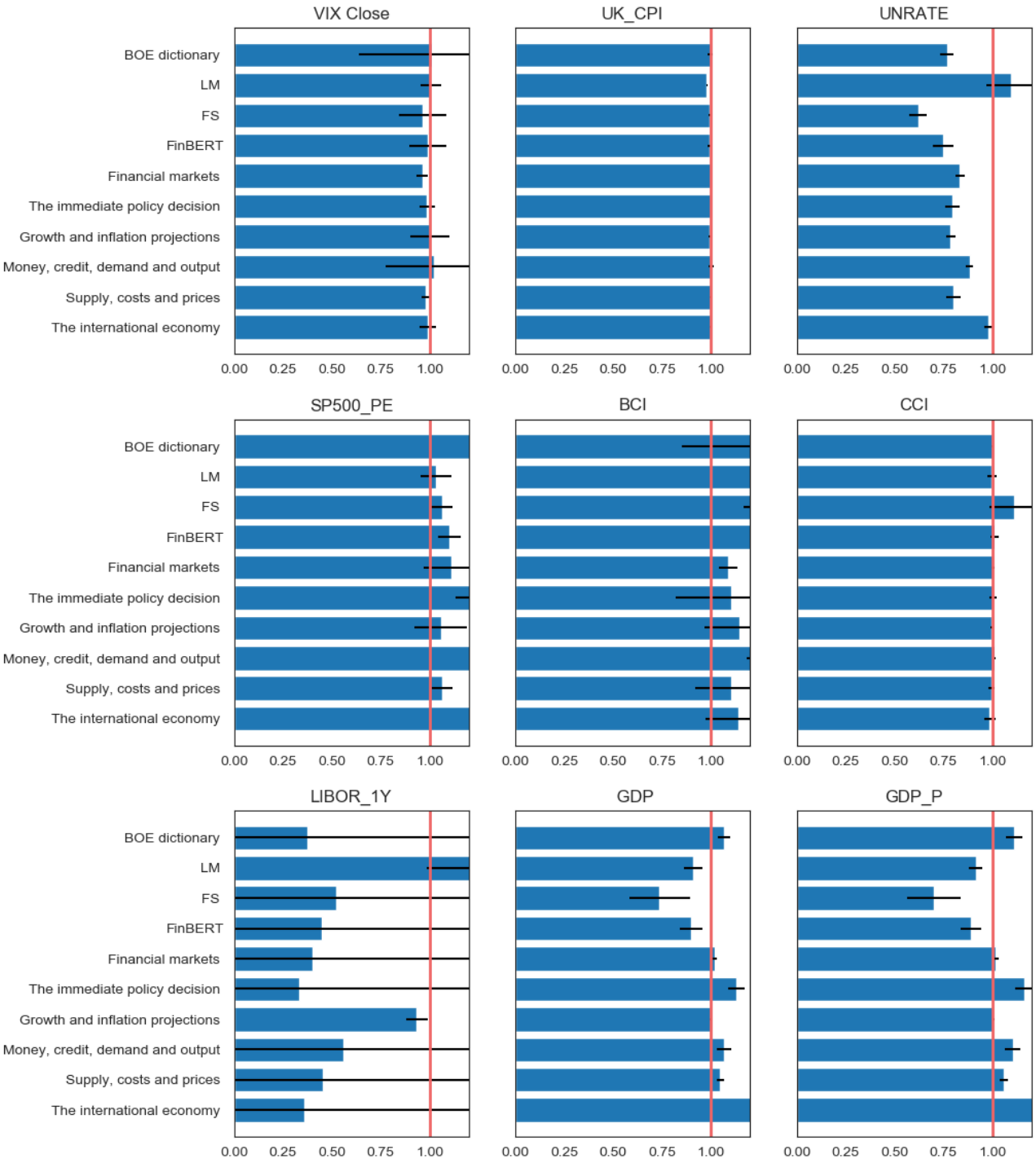

Figure 5.11: Result from the forecasting regression model mentioned in this section. The plot shows the RMSE of a forecasting model with text versus forecasting without a text. The text sentiment it used and compared was stated on the y-axis. And the economical instrument, the model is predicting on, is stated on the title of each subplot. The red line represents the benchmark that the model with or without a new text metrics is having the same RMSE. The bar shows the ratio of the RMSE with text sentiment with horizon = 0, over the RMSE of the Null model. Bars that lie on the left of the benchmark line represented an improvement in forecasting performance with the help of the text sentiment. The confidence interval was the standard deviation over different horizons (0,1,3,6,9 months ahead)

But if we zoom out a little bit, rather than only focus on the interest rate and LIBOR rate, from **Figure 5.11**, we could observe that only very-few target-sentiment pairs are offering significant forecasting powers across all different horizons. And surprisingly, FS dictionary sentiment is performing well in term of predicting GDP and UK unemployment rate. It has only 62.1% of RMSE comparing to the Null model when predicting unemployment rate and 70.0% of RMSE comparing to the Null model when predicting UK GDP growth, both at horizon = 0 (predicting data current month use the data from last month).

This also shows that the central bank documents are carrying information at multiple dimensions. It simply depends on the model we use in order to extract the corresponding information. The FS dictionary was trained on Financial Stability content, hence it is more related to macroeconomic stability, such as unemployment rate and GDP. If we could develop some other dictionaries that include vocabularies focused on other topics, we could extract further information from the central bank documents, instead of the only interest rate.

And actually, the performance of topical sentiment was not that good as it failed to reflect those economical instruments. The reason is probably that the topical sentiment is constructed based on documents labelled by interest rate. If we could find a way to construct a topical sentiment dictionary that not covering interest rate but other financial instruments, the performance might go better. But this will be left for future works.

Finally, because I have observed that the FinBERT and BOE dictionary sentiment have significant predicting power on LIBOR 1Y rate, I run another regression on LIBOR 1Y to look at the coefficients and p-value. And from **Figure 5.12**, we could see that the BOE sentiment are significant with p-value equals 0.001.

	coef	std err	t	P>|t|	[0.025	0.975]
Constant	-0.0145	0.007	-2.002	0.046	-0.029	-0.000
BOE sentiment	0.0976	0.028	3.440	0.001	0.042	0.153
LM sentiment	-0.0144	0.018	-0.782	0.435	-0.051	0.022
FS sentiment	0.0040	0.049	0.080	0.936	-0.093	0.101
FinBERT sentiment	0.1058	0.095	1.110	0.268	-0.082	0.293

Figure 5.12: Running regression on LIBOR 1Y monthly change

From all the evidence I have mentioned above, the BOE dictionary has successfully capture the interest rate sentiment from the central bank documents. And its performance is actually outperforming most of the advanced NLP model in the industry in these specific tasks. And the same methodology can also be applied to other central bank's documents to extract their local sentiment respectively. Also, we have shown that the central bank document carries information in multiple dimensions, we successfully extract GDP and Unemployment sentiment by using FS dictionary. IF we investigate further with new dictionaries, we may capture other macro-economic sentiments that are able to forecast different macro-economic indicators.

Chapter 6

Possible future works

Even though the FinBERT model was under-performed comparing to my dictionary-based model on filtered central bank document, this is largely because of the lack of available training data on economical and interest rate content. And despite this, FinBERT model has already exhibited a relatively amazing sentiment result, where its correlation with LIBOR rate is higher than that of FS dictionary. So if we could Fine-tune the FinBERT model on another 5000 labelled sentences on the topics of interest rate, the performance of FinBERT model would be much better and may even outperform my algorithms proposed in this thesis. Hence, creating such sentence-by-sentence sentiment-labelled central bank dataset could be a possible next step for sentiment extraction on central bank documents.

For example, I split the BOE minutes published from 1997 to 2020 into sentences. Then I hire and distribute these sentences to people from Economic backgrounds and ask them to judge what is the sentiment carried by these sentences, either positive, negative or neutral. Optimally each sentence in assign to 5 different people so that we could have sentiment result with different confidence levels. We could have 100% 80% and 60% agreement level sentences. Then fine-tune the FinBERT model on sentences that have 100%/80% agreement level and from minutes published within the training period (1997 to end of 2009). And then, I apply this FinBERT on all sentences of each document to calculate the sentiment indexes following step 3,4,5 and 6 of the method showed in in Section 5.1.2.

Another possible future work is to rather than applying the dictionary created from BOE documents onto ECB and FED, we could apply **Region-specific Sentiment Algorithm** directly onto ECB and FED documents by replacing the step 1 of document labelling by labelling ECB document based on Euro base rate and labelling FED document by the US base rate. Then we could generate ECB interest rate dictionary and FED interest rate dictionary respective. And hence, we could generate their respective interest rate sentiment index. And we could compare the three dictionaries generated to understand the languages between the different central bank, in order to extract similarities and differences from different regions.

But I am happy with my current thesis and research for now because of the time limit and availability of resources. I wish I could see people working on these possible extensions in the future.

Appendix A

Full vocabulary of the dictionaries

LM dictionary word list

Positive Words				
ABUNDANCE	CHARITY	INDEPENDENT#1	PROFITABLE	WORTH#3
ACCRUE	CIVIL#1	INEXPENSIVE	PROSPER	
ADVANTAGE	COMMUNITY	INHERIT	PROSPERITY	
AFFLUENCE	COMPENSATE	INVALUABLE	PROSPEROUS	
AFFLUENT	COMPENSATION	LEGAL	RALLY	
AFLOAT	CONFEDERATION	LIBERAL#2	RECOMPENSE	
ALLIANCE	CONTRIBUTE#1	LIBERALISM	REINSTATE	
ALLIED	CONTRIBUTION	LIBERTY	REWARD#1	
ALLOWANCE	COOPERATIVE#1	LOYAL	RICH#1	
ALLY#1	COUNCIL	LOYALTY	RICH#3	
ARISTOCRACY	COURTLY	LUCRATIVE	RICH#4	
ARISTOCRAT	CRUSADE	LUXURY	RICH#5	
ARISTOCRATIC	CRUSADER	MERITORIOUS	RICH#6	
ASSOCIATE#1	DONATE	NOBILITY	RICHES	
BACKER	DONATION	NOBLEMAN	RICHNESS	
BARGAIN	ECONOMIZE	NOMINATE	SAVINGS	
BENEFACTOR	ENDOW	PARTNER	SECURITY#2	
BENEFICIARY	ENTREPRENEURIAL	PARTNERSHIP	SKILL#1	
BENEFIT#1	EQUALITY	PATRIOT	SUBSIDIZE	
BENEVOLENCE	EQUITY	PATRIOTIC	SUBSIDY	
BENEVOLENT	FELLOWSHIP	PATRON	SUCCESS	
BEQUEATH	FREEDOM	PATRONAGE	SUCCESSFUL	
BETROTH	FRUGAL	PLEDGE	TACTICS	
BETROTHAL	GAIN#2	PRECIOUS	THRIFT	
BONUS	GENEROSITY	PRICELESS	THRIFTY	
BOOM	GIFT	PRIVILEGED	TREASURE#1	
BREADWINNER	GOLD	PRODUCTIVE	TREATISE	
BUY#2	GUIDE#2	PRODUCTIVITY	TREATY	
CAPITALIZE	HUMANITARIAN	PROFIT#1	UNIMPEACHABLE	
CHARITABLE	INDEPENDENCE	PROFIT#2	VALUABLE	

Negative Words			
ADVERSARY	DEFAULT	JOBLESS	STEAL#2
ANARCHIST	DEFICIT	LAID#2	TARIFF
ANARCHY	DEPRECIATION	LAY#3	TREASON
ANTITRUST	DEPRESSION#2	LIQUIDATE	TREASONOUS
AUTOCRAT	DESTITUTE	LIQUIDATION	TYRANNY
AUTOCRATIC	DICTATE	MISER	UNDERWORLD
BACKWARD	DICTATORIAL	OWE	UNECONOMICAL
BACKWARDNESS	DISCHARGE#1	POLLUTION	UNEMPLOYED
BANISH#1	DOMINATION	POOR#1	UNPROFITABLE
BANISHMENT	ENSLAVE	POOR#2	USURP
BANKRUPT	ENTANGLE	POOR#3	VAGABOND
BANKRUPTCY	ENTANGLEMENT	POOR#4	VAGRANT
BEGGAR	EXPENSE#1	POOR#5	WAR
BLACKMAIL	EXPENSIVE	POVERTY	WARFARE
BRIBE	EXTRAVAGANT	PROPAGANDA	WARLIKE
BROKE#3	FASCIST	RACE#4	WASTE#1
BUM	FEUDAL	RADICAL	WASTE#2
CHEAP	FINE#6	REACTIONARY	
COLD#3	FINE#7	REBELLIOUS	
COLONY	FIRE#2	RECESSION	
COMBAT#2	GAMBLE#1	REFUGEE	
COMMONER	GAMBLE#2	REVOLT	
CONSPIRACY	GHETTO	REVOLUTION	
CORRUPT	HOLE#2	RUIN#1	
COST#1	HUSTLE	SECEDE	
COST#2	HUSTLER	SECESSION	
COSTLINESS	INFLATION	SEGREGATION	
COSTLY	INTERVENTION	SHORTAGE	
CRISIS	INVADE	SIEGE	
DEBTOR	IRON#3	SQUANDER	

Figure A.1: Vocabularies in Loughran and McDonald dictionary

FS dictionary word list														
Positive Words				Negative Words										
able	enhancing	profitable	success	abnormally	contracting	disappointing	eroding	hampered	jeopardised	poorly	sliding	suffering	unresolved	
absorb	enjoy	rallied	successful	abrupt	contraction	discouraging	erosion	hampering	jeopardising	poses	slipped	susceptibility	unrest	
absorbed	excellent	reassuring	successfully	abundant	corrections	disorderly	escalate	headwinds	jeopardize	posing	slowdown	susceptible	unstable	
absorbing	favorable	rebound	upgraded	adverse	costly	disrupt	escalated	hinder	lackluster	problem	slowdowns	tense	unsustainable	
acceptable	favorably	rebounded	upswing	adversely	damaging	disrupted	escalating	hindered	lacklustre	problematic	sluggish	tension	volatile	
achievement	favourable	rebounding	withstanding	aggravate	danger	disruption	escalation	hindering	lagged	problems	sluggishness	tepid	volatility	
adequately	favourably	recouped		aggravated	dangerous	disruptions	exacerbate	hinders	lose	prolonged	slump	threat	vulnerabilities	
alleviated	gain	recover		aggravating	declines	disruptive	exacerbated	hurt	losing	protectionism	slumped	threaten	vulnerability	
alleviating	gained	recovered		aggravation	deep	distortions	exacerbating	illiquid	losses	protracted	slumps	threatened	vulnerable	
beneficial	good	recovering		ailing	deeply	distress	excessive	illiquidity	lost	questions	spill	threatening	weaken	
benefit	grew	recovery		alarming	defaults	distressed	exhausted	imbalance	misalignments	recession	spilled	threats	weakened	
benefiting	grow	regained		anxiety	deficient	distrust	expose	imbalances	misconduct	repercussions	spilling	tough	weakening	
benign	healthy	reopening		arrears	deficits	disturbance	exposed	impair	mispricing	restructure	spillover	troubled	weaker	
better	improve	resilient		bad	delays	disturbances	exposes	impaired	negative	resurfaced	spillovers	tumbling	weakest	
brighter	improved	resolve		burdened	delinquencies	doubts	exposing	impairing	negatively	riskier	spiral	turbulence	weakness	
broaden	improvement	sheltered		challenge	dented	downgrade	fail	impairments	nervousness	setback	squeeze	turbulences	weaknesses	
buoyancy	improvements	smooth		challenges	depress	downgraded	failed	impede	nonperforming	setbacks	squeezed	turbulent	worries	
calm	improves	smoothly		challenging	depressed	downgrades	failing	impeded	overcapacity	severely	stagnant	turmoil	worrisome	
calmed	improving	solid		closure	depressing	downgrading	fails	impediments	overheated	severity	stagnate	unable	worrying	
calming	mitigate	sound		clouded	destabilising	downside	failure	inability	overheating	shaken	stagnated	undermine	worse	
comfortable	mitigated	sounder		compromised	destabilizing	downswing	failures	inadequate	overindebted	shortage	stagnating	undermined	worsen	
confident	mitigates	stabilise		concern	deteriorate	downturn	faltering	ineffective	overvalued	shortages	stagnation	undermining	worsened	
confined	mitigating	stabilised		concerns	deteriorated	downward	fear	inefficient	pessimism	shortfall	strain	underperformance	worsening	
contained	mitigation	stabilising		confronted	deteriorating	drag	fears	insolvencies	pessimistic	shortfalls	strained	underperformed	worst	
effective	opportunity	stabilize		confronting	deterioration	drastic	forced	insolvency	plummeted	shrank	strains	undesirable	writedowns	
efficient	optimism	stabilized		constrain	deteriorations	dropped	fragile	insolvent	plummeting	shrink	stresses	unease		
enabled	outperformed	stabilizing		constrained	detrimental	drying	fragilities	instability	plunge	shrinking	struggle	unemployed		
enabling	positive	stable		constraining	difficult	endanger	fragility	insufficient	plunged	shrunk	struggling	unexpectedly		
enhance	positively	strengthened		contagion	difficulties	erode	gloomy	insufficiently	poor	slid	suffer	unfavorable		
enhanced	preventing	succeeded		contracted	difficulty	eroded	hamper	jeopardise	poorer	slide	suffered	unfavourable		

Figure A.2: Vocabularies in Financial Stability dictionary

Self-created dictionary word list											
Positive Words					Negative Words						
abroad	describ	highest	read	utilis	abil	danger	fear	macro	restructur	troubl
absenc	differenti	idea	recov	variabl	abrupt	declin	feed	markedli	riskier	turbul
accept	doubl	ie	recoveri	withdraw	abund	deep	felt	mervyn	save	turmoil
access	east	implement	reform	worker	address	default	financ	mortgag	saw	undermin
accommod	effici	improv	resolv		adopt	dent	food	neg	sentiment	undesir
age	element	instead	restrain		advers	deterior	fragil	nervous	setback	unsustain
aris	enjoy	interpret	sensit		aggrav	difficult	gfk	notic	sever	upon
arrang	equilibrium	latter	shortag		alon	difficulti	gloomi	origin	sheet	vulner
aspect	evolv	leav	singl		alongsid	disrupt	hard	outcom	shortfal	weak
attribut	examin	life	skill		awar	distress	head	overh	shrink	weaken
banker	exceed	link	smooth		away	downgrad	illiquid	persist	situat	weaker
basic	excel	mainli	social		bad	downsid	illustr	pessim	slow	weakest
behaviour	expenditur	mitig	someth		bear	downsw	imbal	pessimist	slowdown	wholesal
benefit	fail	narrow	spare		becam	downturn	imf	petrol	sluggish	widen
benign	fast	offici	specif		boom	downward	impair	poor	sometim	widespread
better	favour	optim	specul		challeng	drag	imped	precis	spiral	worri
book	fed	outperform	spot		charter	dri	inadequ	problem	squeez	wors
broad	foreign	paid	spring		collaps	driven	insolv	problemat	stabilis	worsen
buy	furthermor	particip	stabl		collater	drop	introduct	prolong	stephen	worst
centuri	ga	partli	strengthen		commerci	ecb	inventori	pronounc	stimul	
china	gain	pension	tax		consensu	elsewher	judgement	properti	strain	
complet	gap	pick	technic		constraint	ensur	justifi	protract	stress	
complic	gather	pickup	threat		contagion	erod	king	question	struggl	
conduct	gave	pmi	transfer		contract	event	lag	react	suffer	
consecut	german	posit	twenti		correct	exacerb	learn	recess	summer	
constant	good	post	unanim		costli	excess	liquid	recognis	surprisingli	
contain	group	prevent	unchang		crisi	expo	lose	reinforc	task	
curv	grow	primarili	unexpect		cut	extrem	loss	remark	tension	
cyclic	grown	principl	upsw		damag	failur	lost	resili	threaten	
date	healthi	profil	upward		dampen	falter	lowest	respond	treasuri	

Figure A.3: Vocabularies in self-created dictionary after language filtering

Appendix B

LDA Model details

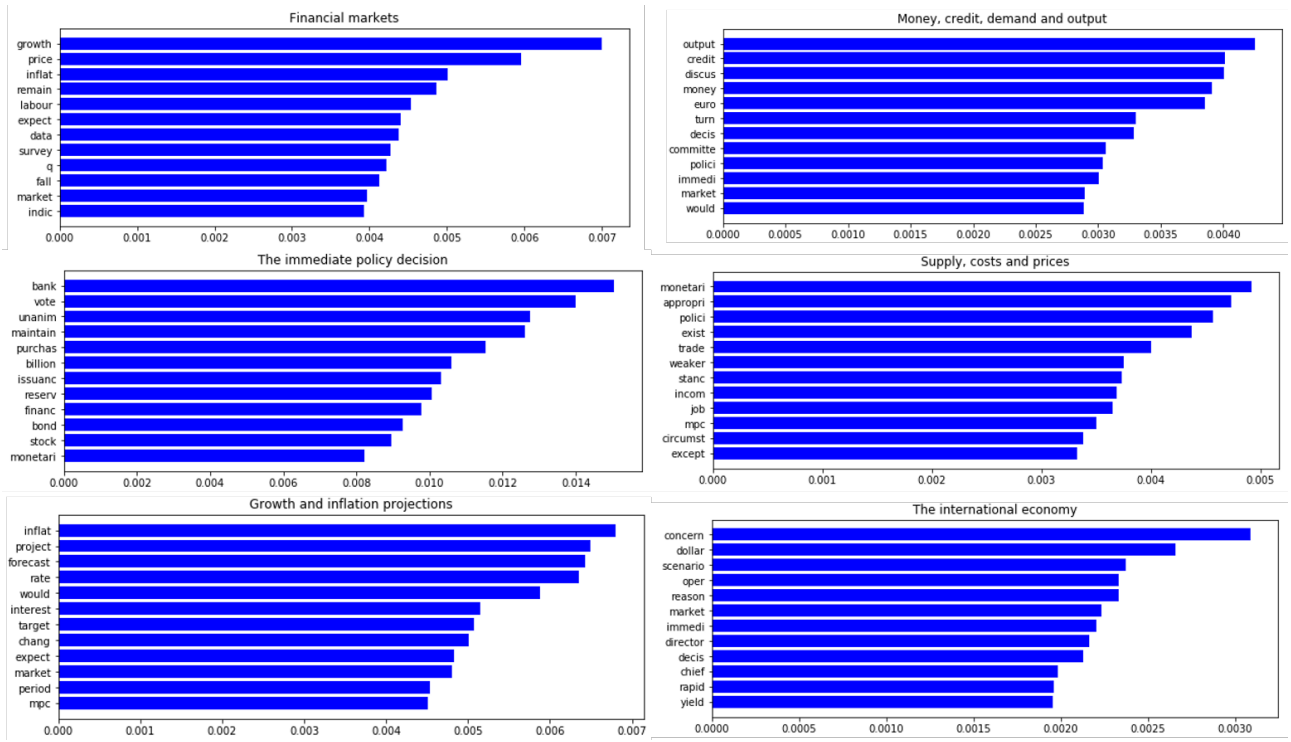

Figure B.1: LDA model words' importance and their respective topics

List of Figures

Bibliography

[1] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models, 2019.

[2] Andrew Y.; Jordan Michael I Blei, David M.; Ng. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] Bing Liu. Sentiment analysis and opinion mining. volume 5, 05 2012.

[4] Basant Agarwal and Namita Mittal. *Machine Learning Approach for Sentiment Analysis*, pages 193–208. 01 2014.

[5] Abinash Tripathy, Ankit Agrawal, and Santanu Rath. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, 57, 03 2016.

[6] Ricardo Correa, Keshav Garud, Juan M. Londono, and Nathan Mislang. Sentiment in Central Banks' Financial Stability Reports. International Finance Discussion Papers 1203, Board of Governors of the Federal Reserve System (U.S.), March 2017.

[7] Tim Loughran and Bill Mcdonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66:35 – 65, 02 2011.

[8] Aliaksei Severyn and Alessandro Moschitti. Twitter sentiment analysis with deep convolutional neural networks. pages 959–962, 08 2015.

[9] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis : A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 01 2018.

[10] Zheng Ke, Bryan Kelly, and Dacheng Xiu. Predicting returns with text data. *SSRN Electronic Journal*, 01 2019.

[11] Ellen Tobback, Stefano Nardelli, and David Martens. Between hawks and doves: measuring central bank communication. *ECB Working Paper Series*, 07 2017.

[12] Maik Schmeling and Christian Wagner. Does Central Bank Tone Move Asset Prices? CEPR Discussion Papers 13490, C.E.P.R. Discussion Papers, January 2019.

[13] Eleni Kalamara, Arthur Turrell, Chris Redl, George Kapetanios, and Sujit Kapadia. Making text count: economic forecasting using newspaper text. Bank of England working papers 865, Bank of England, 2020.

[14] Saskia ter Ellen, Vegard H. Larsen, and Leif Anders Thorsrud. Narrative monetary policy surprises and the media. Working Papers No 06/2019, Centre for Applied Macro- and Petroleum economics (CAMP), BI Norwegian Business School, October 2019.

[15] David O. Lucca and Francesco Trebbi. Measuring Central Bank Communication: An Automated Approach with Application to FOMC Statements. NBER Working Papers 15367, National Bureau of Economic Research, Inc, September 2009.

[16] Ellen Tobback, Stefano Nardelli, and David Martens. Between hawks and doves: measuring central bank communication. Working Paper Series 2085, European Central Bank, July 2017.

[17] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78, 11 2018.

[18] Hemant Misra, Olivier Cappé, and François Yvon. Using lda to detect semantically incoherent documents. *CoNLL 2008 - Proceedings of the Twelfth Conference on Computational Natural Language Learning*, 08 2008.

[19] Binyam Gebre, Marcos Zampieri, Peter Wittenburg, and Tom Heskes. Improving native language identification with tf-idf weighting. 01 2013.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[22] Pekka Malo, Ankur Sinha, Pyry Takala, Pekka Korhonen, and Jyrki Wallenius. Good debt or bad debt: Detecting semantic orientations in economic texts, 2013.

[23] Jeffrey Pennington, Richard Socher, and Christoper Manning. Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01 2014.

[24] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

[25] Srikumar Krishnamoorthy. Sentiment analysis of financial news articles using performance indicators. *Knowledge and Information Systems*, 56(2):373–394, Nov 2017.

[26] Macedo Maia, Andr Freitas, and Siegfried Handschuh. Finsslx: A sentiment analysis model for the financial domain using text simplification. pages 318–319, 01 2018.

[27] Scott Baker, Nicholas Bloom, and Steven Davis. Measuring economic policy uncertainty. *The Quarterly Journal of Economics*, 131:qjw024, 07 2016.