**Daniel H. VEDIA-JEREZ**

**Master Dissertation Proposal**

**A NATURAL LANGUAGE PROCESSING PIPELINE TO ANALYZE CENTRAL BANKS POLICY CHANGES**

**Introduction**

This document describes the proposal for the Master Dissertation, it is focused on analyzing Central Banks policy changes by means of Natural Language Processing (NLP) techniques.

**Business Context**

Central Banks have regular meetings of their open market committees to determine the monetary policy. At each meeting, it publishes press conference minutes, statements as well as scripts in the website. In addition to this regular meetings, the members’ speeches and testimonies are also scripted on their respective websites. At a meeting, the policy makers discuss, vote and decide the monetary policy and publish the decision along with their view on current economic situation and forecast. The Central Banks intend to indicate their potential future monetary policy in their publications as a measure of market communication.

The objective of this project is to find latent features in those texts published by Central Banks. First, we intent to apply machine learning to economic indices to see the performance of prediction on those numerical data. Then, add pre-processed text data as an additional feature to see if it contains meaningful information. Finally, to apply Deep Learning techniques such as LSTM/RNN and BERT to see if these can better predict the rate hike/lower at each Central Banks committee meetings.

**1. Retrieving Market**

Economic data as interest rates and major economic indices can be obtained from Central Banks statistical website, the variables included in the document are: Central Banks interest rate, GDP, CPI/PCE[[1]](#footnote-2), Unemployment, Retail Sales and Home Sales, Manufacturing PMI and Service PMI(formerly known as “Non-Manufacturing Index or NMI). Daily Treasury yield rates.

Also we could explore the details of the data on each website and downloaded, sometimes it’s much more convenient to use Quandl, which provides Web APIs and Libraries to retrieve all the data in the same manner[[2]](#footnote-3). Once you create an Quandl Account, API Key is provided.

Regarding the documents from Open Market Committeess we use the data sources from the each respective Central Bank website and the Basel Committee on Banking Supervision (BIS), the latest stores Central Banks speeches but not all of them, so is necessary to use Central Banks websites to gather all the documentation.

**3. Preliminary Analysis-**

In order to see if the documents may contain some useful insight to predict Central Banks rate, we use Loughran and McDonald Sentiment Word List to measure the sentiment of statement. This dictionary contains several thousands words appearing in financial documents such as 10K, 10Q and earnings calls categorized to positive, negative, etc. It includes words in different forms, so stemming or lemmatising should not be applied. We will consider a simply technique to flip the sentiment for negation (e.g. can’t, isn’t, no).

Also checking the moving average of net sentiment with actual Central Banks rate decisions at each Committee meeting. It is expected to get a certain correlation with Central Banks target rate, but it will not be easy to see it during the Financial Crisis where the rate was at Effective Lower Boundary[[3]](#footnote-4) and Quantitative Easing (QE) was taken place. We will treat QE announcement as a lowering rate event.

**4. Feature engineering: Taylor rules**

As a part of feature engineering, we will calculate the Taylor rules[[4]](#footnote-5), Balanced-approach Rule, and Inertial Rule from raw data for each Central Bank and see whether the first derivatives and difference from Central Banks rate could be used.

**5. Pre-processing Text Data**

We expected to get around 200 decisions on monetary policy over the last two decades and a half. Depending on the models some inputs cannot be used due to missing data or available timing limitations.

One of common issues you may face during text processing is how to handle long text in machine learning. Most of the neural net based algorithms are not capable to analyzing such long texts like 10,000 words — 500 at maximum. Most of our input texts are too long to analyze as a whole document.

A typical solutions to this problem is either to use other algorithms such as the Jaccard/Cosine similarity on the document vectors or to find a way to split the long text or to use some techniques to shorten such as text summarisation. To deal with this issue we propose a technique to split the text by the number of words (i.e. 200 words with overlapping of 50 words) as shown below. This is a simple automated way but easily lose the context because the extracted 200 words may or may not contain relevant text and even off the topic.

Another issues is data imbalance — in our case, we expect a “hold” rate decision (no change on policy) for more than 50% chance and available decisions are around 200. Without having enough data, machine learning can easily over-fit the training data.

**6. The Machine Learning Models**

At high-level, the model takes textual inputs and meta inputs to predict three classes: Raise, Hold or Lower as follows. The point is how to combine textual input with numerical inputs and there are different ways to implement it.

Here we propose to implement the following six models in addition to the baseline model.

1. ***Baseline Model***

This model does not use textual inputs but just take the meta inputs. We then compare almost 14 different classifiers with their default parameters to grab first the baseline performance quickly. Then, apply RandomizedSearchCV and GridSearchCV to find optimal hyper parameters. StratifiedKFold is used for the cross validation.

1. **Cosine Similarity**

The texts are vectorised by the *Tfidf* using *Loughran-McDonald dictionary*, which is used in preliminary analysis, and calculate the cosine similarity between two consecutive meetings. This value is the degree of change in the text direction (i.e. cosine of vectors), which may indicate possible policy changes. This is then combined with the economic indices used in the baseline model.

1. **Tfidf**

Instead of using the cosine similarity, only use the Tfidf vector itself as input. This would only work if the Tfidf vector directly holds meaningful information on the rate change. Using the tokenized text in the previous step, concatenate the Tfidf Vector with Non-textual inputs using the FunctionTransfomer.

1. **Long Short-Term Memory (LSTM)**

LSTM is a popular Recurrent Neural Network architecture that can hold long-term memory and short-term memory for sequence learning. Although there are a number of improved versions such as bidirectional LSTM, the one used here is a simple plain model.

The output from the deep neural network is combined with economic indices after dropout and dense layer. Then, separate the input to textual data and numeric data. The data loader is customized to yield the two types inputs. Training process is the same as usual case of processing text by LSTM.

1. **LSTM+GloVe (Global Vectors for Word Representation)**

The previous LSTM model creates own word embedding but there’s also pre-trained embedding. Here we plan to use GloVe which was trained by Wikipedia and Gigaword (6B tokens). The idea is that the pre-trained word representation would boost the performance of the randomly initialized model.

The training steps are the same as the previous LSTM model.

1. **Bidirectional Encoder Representations from Transformers (BERT)**

BERT is a transformer based language model as opposed to RNN. Apart from the model and BERT’s own tokenisation, the rest of architecture stays the same as the LSTM based model above. Usually it would be sufficient to use *transformers.BertForSequenceClassification* for the model but here needs to create own definition to concatenate text with non-text inputs.

1. **BERT + Pre-Sentiment Analysis**

Finally, we pretend to take another approach — instead of training the model directly on Central Banks Committees text, first we train the model on other financial texts for sentiment analysis task. Then used the trained BERT model to analyze the sentiment of each sentence in Central Banks Committees text and calculate sentiment scores, which were then aggregated for each document and used as inputs to another ML model to predict the Central Banks Committee's decision.

**7. Results**

The following shows the comparison of scores between the tested models. The deep neural network models performed poorly, which is basically due to lack of enough data to train these complex models. The last model with pre-sentiment analysis outperformed the other models but need to examine further if this improvement is with meaningful significance and consistency, and not due to just any additional inputs.

**Conclusion**

We examined whether Central Banks Committeesstext data contains useful insight to predict the Central Banks target rate decision (i.e. Raise, Hold or Lower) at the next Central Banks Committeessmeeting. We could observe some useful information in the text to predict Central Banks Committeessdecision better. However, we could not improve the text based prediction performance by Neural Network. This is partly because there are small number of test data to train with each text very long. As a future work, there’re two main areas to improve:

1. Tackle the lack of enough training data — The models have clearly overfitted to train samples and failed to generalise well, especially boosting algorithms are prone to overfitting. Hyperparameter tuning and imputation was considered there. In addition, configuring the model and splitting data to augment the training data by synthetic approach could potentially beneficial.

2. Improve input text quality — The input texts contain a lot of irrelevant paragraphs, which have nothing to to with Central Banks target rate decision. For example, there are information about regulations, organisation structure and infrastructures. Filtering out less relevant inputs will improve the accuracy of the model prediction as well as training efficiency.

Note that the code in this post is just some extracts. Please refer to Github repo for the complete source code.

1. There are two common measures of inflation in the US today: the Consumer Price Index (CPI) released by the Bureau of Labor Statistics and the Personal Consumption Expenditures price index (PCE) issued by the Bureau of Economic Analysis. The CPI probably gets more press, in that it is used to adjust social security payments and is also the reference rate for some financial contracts, such as Treasury Inflation Protected Securities (TIPS) and inflation swaps. The Central Bankseral Reserve, however, states its goal for inflation in terms of the PCE. [↑](#footnote-ref-2)
2. All the data above are publicly available and free for personal use but you should always check the license terms in the original source in accordance to your objective. [↑](#footnote-ref-3)
3. Zero-bound is an expansionary monetary policy tool where a central bank lowers short-term interest rates to zero, if needed, to stimulate the economy. A central bank that is forced to enact this policy must also pursue other, often unconventional, methods of stimulus to resuscitate the economy. [↑](#footnote-ref-4)
4. The Taylor rule is one kind of targeting monetary policy used by central banks, as a central bank technique to stabilize economic activity by setting an interest rate. The rule is based on three main indicators: the federal funds rate, the price level and the changes in real income. The Taylor rule prescribes economic activity regulation by choosing the federal funds rate based on the inflation gap between desired (targeted) inflation rate and actual inflation rate; and the output gap between the actual and natural level. [↑](#footnote-ref-5)