Daniel H. VEDIA-JEREZ

Master Dissertation Proposal

A NATURAL LANGUAGE PROCESSING PIPELINE TO ANALYZE CENTRAL BANKS POLICY CHANGES

1. Introduction

This document describes the proposal for the Master Dissertation, it is focused on analyzing Central Banks policy changes employing Natural Language Processing (NLP) techniques. We plan to examine whether Central Banks Committees text data contains useful insight to predict the Central Banks target rate decision (i.e., Raise, Hold, or Lower).

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 on the website. In addition to these regular meetings, the members' speeches and testimonies are also scripted on their respective websites. At a meeting, the policymakers discuss, vote, and decide the monetary policy and publish the decision along with their view on the 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. In that way, we focused in the most important world Central Banks (Federal Reserve, European Central Bank, Bank of England and Bank of Japan) from 2009 to the first quarter of 2021.

First, we intend 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 meeting.

2. Retrieving market data

Economic data as interest rates and major economic indices can be obtained from the Central Banks statistical website, the variables included in the document are Central Banks interest rate, GDP, CPI/PCE[1], unemployment, retail sales and home sales, manufacturing PMI, service PMI (formerly known as "Non-Manufacturing Index or NMI), and Treasury yield rates.

---

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

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]. Once you create a Quandl Account, API Key is provided.

Regarding the text documents from the Open Market Committees, we use the following sources from 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-**

To see if the documents may contain some useful insight to predict the Central Banks rate, we use *Loughran and McDonald Sentiment Word List* to measure the sentiment of the statement. This dictionary contains several thousand words appearing in financial documents such as 10K, 10Q and earnings call categorized too positive, negative, etc. It includes words in different forms, so stemming or lemmatizing should not be applied. We will consider a simple 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] and Quantitative Easing (QE) was taken place. We will treat the QE announcement as a lowering rate event.

4. **Feature engineering: Taylor rules**

As a part of feature engineering and to add additional evidence to our work, we will calculate the Taylor rules[4], 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.

---

also the reference rate for some financial contracts, such as Treasury Inflation Protected Securities (TIPS) and inflation swaps. The Central Banks however, states its goal for inflation in terms of the PCE.

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.

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.

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.

5. **Pre-processing Text Data**

We expected to get around 600 decisions on monetary policy over the last decade. Depending on the models some inputs cannot be used due to missing data or available timing limitations.

One of the 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 of 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 solution 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 summarization. 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 loses the context because the extracted 200 words may or may not contain relevant text and even off the topic.

Another issue 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 600. Without having enough data, machine learning can easily over-fit the training data. To tackle the possible lack of training data — we could split data to augment the training data by synthetic approach that could potentially benefit the model. In addition, we could avoid the use of boosting algorithms that are prone to overfitting train samples and failed to generalize well.

Finally, to improve input text quality — the input texts contain a lot of irrelevant paragraphs, which have nothing to do with Central Banks' target rate decision, for example, there is information about regulations, organization structure, and infrastructures. Filtering out less relevant inputs will improve the accuracy of the model prediction as well as training efficiency.

6. The Machine Learning Models

At the 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.

a) ***Baseline Model***

This model does not use textual inputs but just takes 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 hyperparameters. StratifiedKFold is used for cross-validation.

b) **Cosine Similarity**

The texts are vectorized by the *Tfidf* using the *Loughran-McDonald dictionary*, which is used in the preliminary analysis and calculates 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.

c) **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.

d) **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 few 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 into textual data and numeric data. The data loader is customized to yield the two types of inputs. The training process is the same as the usual case of processing text by LSTM.

e) **LSTM+GloVe (Global Vectors for Word Representation)**

The previous LSTM model creates its 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.

f) **Bidirectional Encoder Representations from Transformers (BERT)**

BERT is a transformer-based language model as opposed to RNN. Apart from the model and BERT's tokenization, the rest of the 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.

g) **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.