ECO480 Term Paper

Yiyu Li, Jasmeet Ajrha
2023-06-25

Research Question

How does the tone of Bank of Canada (BoC) announcements correlate with changes in prices in the S&P/TSX composite index?

Using a language model for sentiment analysis to make predictions on whether the announcements given by the Bank of Canada are rather positive or rather negative, we want to determine if these predicted sentiments correlate with changes in asset prices in the S&P/TSX composite index, within 24 hours of the announcement.

Background Information

According to Ben Bernanke, former chairman of the US Federal Reserve (USA's central bank equivalent to the BoC), "monetary policy is 98% talk and 2% action"[1]. When central banks release data or take policy action, they usually make announcements regarding it using carefully chosen words. They use this careful communication to explain what they're doing, how they think the economy is doing, and what they expect to happen in the future. People in the market pay close attention to what the central bank says, not just to the content, but also to the tone of how it is said. Central bank announcements aim to converge the market's expectations of the future economic outlook to a particular monetary goal. The central bank's narrative, curated by the sentiment of their announcements, is therefore carefully written and is used as a monetary tool to advance market expectations.

Investors determine the value of stocks by looking at two main factors: their predictions of future cash flows generated from owning the stocks, and their predictions of future discount rates, which represent the expected returns for investing in those stocks. These forecasts are influenced by various factors, including the investors' expectations of the central bank's future monetary policy actions and the economic outcomes resulting from those actions.

A question that comes up is whether and how the tone of central bank communication affects the price of assets. If we find that asset prices do respond to the announcements by the BoC, it could signify that financial markets trust that the central bank's future actions will reflect the tone of their announcement, hinting at the bank's credibility to the market. By

[1] Schmeling et al., (2019)

analyzing how BoC tone affects asset prices, we can also make predictions on potential trading signals ahead of the market.

Past literature can guide us on what results we might expect to find from our research question. In the Canadian market, a study by Hayo et al. (2010) show that BOC announcements exert a significant effect on Canadian financial market returns. This leads us to suspect that the S&P/TSX composite index might show a response to the tone of BoC announcements. Many financial news outlets also tend to report changes in the S&P/TSX composite index following BoC policy announcements[2].

In the European market, another study by Schmeling et al. (2019) wanted to find how central bank tone affects asset prices. Their study looked at press conferences held by the European Central Bank (ECB) from 1999 to 2017. After controlling for the bank's policy decisions and economic factors, the researchers found that the way the central bank communicates influences the price of assets. These findings support the notion that the tone in the communication used by central banks can be used as a tool for monetary policy. It has the potential power to impact how willing market participants are to take risks in the stock market.

The efficient market hypothesis argues that an announcement's effect on asset prices should only exist if interest rate announcements are "news", or in other words, a surprise. Stock prices at any given time reflect all available information, so that only new information should move stock prices. If information was available that made it seem highly likely that the BoC will change the key policy rate in the future, investors of the asset will respond such that the prices at that time will have adjusted. The efficient market hypothesis shapes the design of our approach to the research question. Looking at a past study, Rosa et al. (2008) asked what the degree of efficiency was of the Euribor futures market when analyzing the impact of European Central Bank (ECB) announcements to its asset prices. Their results suggested that the Euribor futures market incorporated the news from the ECB policy decisions very quickly, in less than five minutes, and the news from the ECB President's speech in around one hour. These results suggest we look at only a 24-hour window before and after the policy to capture any sharp discontinuity in asset prices correlating to the BoC announcement.

A Proof of Concept

This paper will be a proof of concept for determining how to best approach answering our research question. Our main limitation in this project is that we do not have specific training data readily available. For the purpose of this project, we will be creating a dataset with

[2] Following the June 7, 2023 BoC announcement of a 25 basis point rate hike, the S&P/TSX composite index fell by 0.4% at the close of the day. "Today's message that more hikes are coming is resetting many peak terminal rate bets", Edward Moya, a senior market analyst at OANDA said in a note to Reuters. (Smith, Fergal).

manually recorded sentiment labels on the BoC announcements. Because of limited time, we will only have labelled a small collection of BoC announcements. We approach the challenge of having limited training data by using another readily available larger dataset containing pre-labelled sentiments (positive, negative, neutral) on sentences from a collection of financial phrases. We will use the financial text to train the model.
We are hoping the financial text will have enough similarity with the BoC text to allow the model to make predictions of BoC sentiments. However, we understand that using a different training set from the target dataset can potentially lead our model to give non-robust predictions. Therefore, we also choose to focus on different word embedding techniques as the input layer in the network model to determine how to best optimize the LSTM model's predictions.

Data

Data Sources

This study involves 3 data sources:

Unprocessed Dataset 1: Financial Phrase-bank Dataset [1]
This dataset consists of 4840 sentences from financial news in English, categorized by sentiment (positive, negative, neutral). There was a 75% agreement rate of the sentiment labels given to each sentence, from 5-8 annotators.
[Source: reference 10]

Unprocessed Dataset 2: Central Bank Publications Dataset [2]
This dataset consists of 1000+ dated and titled central bank press releases and speeches from the Bank of Canada website, from 2010 to the present.
[Bank of Canada press releases and speeches. Retrieved from https://www.bankofcanada.ca/press/speeches/?mt_orderby=0&content_type%5B%5D=4&mtf_date_after=2010-01-01]

Unprocessed Dataset 3: TSX Composite Index Dataset [3]
This dataset contains the opening and closing prices of the S&P/TSX stock index, enabling examination of the bank announcement's sentiment impact on asset prices.

[S&P/TSX Composite index Historical Data. Yahoo Finance. Retrieved from https://finance.yahoo.com/quote/%5EGSPTSE/history?period1=883612800&period2=1685923200&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true]

Data Collection Methodology

Dataset 2 is collected specifically for this study. Its collection involves iterating through BoC web pages, obtaining the links to each announcement from the page, then extracting contents from the linked announcements. These steps are taken on Python programming software using keyword searching methods from BeautifulSoup4 to locate the links and article contents. The complete code can be found in the **Appendix**.

The collection process for Dataset 2A further involves selecting articles published by the Bank of Canada with relatively polarized titles and assigning sentiment labels to each paragraph within the article. The sentiment labels ('positive', 'negative', or 'neutral') are determined after careful review of each paragraph. The labelled paragraphs are then aggregated to find the dominant sentiment of the announcement by a majority vote. This process yields a labeled dataset for 20 BoC announcements. Because our project is a proof of concept for how to best approach our research question, we target polarized titles to help understand the scope of our chosen model's ability to predict those sentiments, and later to understand the scope of the stock market's response to those polarized announcements.

Data Preprocessing

Dataset 1 and Dataset 2 is preprocessed to ensure the format of the target and test datasets were consistent and suitable for use in the chosen model. For both Datasets, the text is lowercased and lemmatized, and unwanted stop-words are removed by referencing a dictionary of stop-word expressions. Tokenization is applied to vectorize the text and convert it into sequences. Padding is implemented to maintain a uniform sequence length, as required for inputting into the modeling.

Preprocessing Dataset 3 involves calculating the percentage change in stock prices within 24 hours before and after central bank publications.

Modelling

To ultimately analyze the impact of BoC announcement sentiments on asset prices, our project involves three steps:

Step 1: Sentiment Analysis using Word Embeddings in an LSTM Model

There are two factors to consider in text analysis:

- Semantic relationship between words: different words may mean the same thing (synonyms). To capture their semantic relationship, words will be mapped to numerical vector representations, so that words that have closer meanings will have closer numerical values.
- Sequential dependencies within a text: The meaning of sentences vary according to the ordering of words in the sentence. Using a sequential model should help to recognize these patterns within the text data.

Models that capture the semantic and sequential understanding of text are extremely well implemented.

We employ the LSTM (Long Short-Term Memory) model to perform sequential analysis on the training data from Dataset 1.

We go through three different approaches for embedding numerical representation of text to capture semantics: an embedding layer in TensorFlow trained on Dataset 1, Word2Vec skip-gram embeddings pre-trained on Wikipedia text, and manually tuned Word2Vec CBOW (continuous bag-of-words) embeddings trained on Dataset 1 and Dataset 2.

Why we Choose LSTM

With its ability to capture long-term dependencies and contextual information, the LSTM (Long Short-Term Memory) model is well-suited for predicting sentiments from sequential text data (Hochreiter 1997). Such data includes financial news headlines (Dataset 1) and central bank announcements (Dataset 2), which often contain extensive paragraphs with nuanced language and complex financial information. The LSTM model excels at understanding the context of the entire paragraph by retaining and utilizing information from earlier sentences, ensuring that the sentiment prediction is not solely based on the current sentence.

Initial Model Construction

Our initial base model is a sequential model consisting of 3 layers:

```
# build the LSTM model

# model structure:

# sequential model
model = Sequential()
# embedding layer
model.add(Embedding(input_dim=vocab_size, output_dim=32,
input_length=max_len))
# LSTM layer
model.add(LSTM(units=32))
# dense layer for classification

# choice of activation function: softmax
# common choice for multi-class classification
# it outputs a probability distribution across the sentiments
(pos/neg/neutral),
# ensuring that the predicted probabilities sum up to 1.
model.add(Dense(units=3, activation='softmax'))

# Specify the loss function, optimizer, and evaluation metrics

# choice of loss: categorical_crossentropy
# common choice for multi-class classification
model.compile(loss='categorical_crossentropy',
              optimizer='adam', metrics=['accuracy'])


# Define early stopping callback
early_stopping = callbacks.EarlyStopping(patience=5,
restore_best_weights=True)

# train the model
# epochs and batch size can be tuned later on
model.fit(X_train, y_train, validation_data=(X_val, y_val),
          epochs=30, batch_size=128, callbacks=[early_stopping])
```

- A trainable embedding layer in TensorFlow. This layer is trained along with the rest of the model during the training process, within which it learns the sentiment of words from scratch based on Dataset 1. We choose 32 units in the input layer of our neural network after observing the training and validation accuracy performance from inputs

with 16 to 128 units. This choice mitigates the overfitting problem while retaining enough contextual information.
- A LSTM layer, which processes the input sequence and captures long-term dependencies.
- A dense layer for 3-class classification using softmax activation, for providing a probability distribution over the 3 classes (positive, neutral, negative). The output layer will show the probabilities that represent the likelihood of each class being the correct output.

Additionally, an early stopping callback is defined to monitor the validation loss and stop the training process if there is no improvement in performance on the validation set after 5 epochs.

In summary, the objective of our embeddings is to find a numerical representation of words that encodes the semantic proximity between words, to then input into the LSTM model. The LSTM model then captures the sequential understanding of words through its ability to capture long-term dependencies in sequential data.

Initial Model Estimation

The training-test-validation ratio is 64:20:16. The base model achieves a training accuracy of 90% and a test accuracy of 80%.

However, the predictive accuracy of the trained network drops substantially to 40% when applied to our target Dataset 2, which contains the central bank announcements. We identify three potential reasons for this discrepancy in predictive accuracy:

1. Reason 1: The model overfits on the training data from Dataset 1
2. Reason 2: There is an unbalanced sentiment distribution in Dataset 1 (6:1:3, neutral:negative:positive)
3. Reason 3: The TensorFlow self-trained word embeddings on Dataset 1 do not transfer well to the BoC announcement domain

To optimize the performance of our model, we address these issues in **Step 2**.

Step 2: Model Optimization

In this step, we focus on optimizing the base model from Step 1 to address the identified issues. To tackle reasons 1 and 2, we construct 5 variants of the base model, each incorporating reweighting, resampling, or aggregation techniques. Below is a brief summary for each model. The complete code, including detailed comments for model construction and testing, can be found in the appendix.

Model 1: switch to custom loss functions that penalize incorrect predictions on the minority classes ('negative') more heavily, aiming to account for the unbalanced sentiment distribution.

Model 2: introduce class weights during model training to account for the unbalanced sentiment distribution. Weights are inversely proportional to the class frequencies.

Model 3: under-sample the majority class ('neutral') and oversample the minority classes ('negative') to account for the unbalanced sentiment distribution.

Model 4: mix 20% of Dataset 2A into the resampled data from Model 3, then train the model with mixed data and test on both the test data from Dataset 1 and the rest 80% of Dataset 2A.

Model 5: an ensemble of Model 1-4 by majority voting across their predicted labels.

Model 1-5 Estimation

The test accuracy for all models fluctuates around 40%. The confusion matrices of Models 1 and 2 indicate a strong bias towards predicting 'neutral' or 'positive' sentiments, while the true 'negative' rate remains extremely low. On the other hand, Model 3,4,5 successfully addresses the unbalanced sentiment problem by producing a similar ratio of each sentiment label. However, the true class rate remains low.

Modeling with Pre-Trained Word2Vec Embeddings

To address Reason 3, we explore an alternative approach by replacing the TensorFlow embeddings with pretrained Word2Vec embeddings. Pretrained embeddings leverage large-scale unsupervised training to capture broader semantic information, making them more effective in capturing long-term dependencies. Mikolov's paper on the training of the Word2Vec model highlights the benefits of pretrained embeddings, particularly when trained on extensive and diverse text. The model's embedding for a word not seen in its pretrained data is possible based on the embedding of similar words in its training corpus.

Word2Vec is an embedding neural network that uses skip-gram to predict the context from a word. We use a pretrained Word2Vec from Tensorflow hub, which uses Skip-Gram architecture to create a 250-dimensional vector from training on Wikipedia text. Word2Vec gives low dimensional, high-quality embedding that gives the LSTM model a better-quality input to train on than the trainable embedding layer from TensorFlow we used earlier.

After converting the entire Dataset 1 text to 250-dimensional numerical vectors using the pre-trained Word2Vec embedding model. The LSTM layer again has 32 units to mitigate the

overfitting problem. The learning rate on the Adam (Adaptive Moment Estimation) Optimization is set to 0.005. Since we are working with a small training dataset, using a lower learning rate can help prevent overfitting. The model will be less influenced by noisy data in the training set since it is making smaller updates towards optimal convergence.

We consider 5 different models with pretrained Word2Vec embeddings.

- **Model 6:** We train the LSTM model with the full Dataset 1 (no training nor validation set). This makes the model more generalizable at the cost of not being able to measure the level of overfitting (with no validation nor test set from Dataset 1).

The next 4 models apply different regularization techniques on Model 6 to determine how to optimally prevent overfitting:

- **Model 7:** Regularize by applying batch normalization. This standardizes the intermediate input to maintain a consistent distribution of inputs to each layer for stabilizing the training process.

- **Model 8:** Switch from training on full dataset to training-test split. Same modeling approach is applied.

- **Model 9**: the same mixed-data approach in Model 4 is applied on Model 6.

- **Model 10:** Regularize by applying 10% dropout in the LSTM layer. This randomly sets 10% of the layer's input to 0 to prevent overfitting.

Modeling with Own Estimated Word2Vec Embeddings Trained on Datasets 1 and 2

By training the embedding model directly on the target dataset, we hope to pinpoint whether a low predictive accuracy would come from poor quality semantic embeddings, or from poor sequence modelling from the LSTM model.

Using Python's gensim library, we implement a Word2Vec algorithm with CBOW (continuous bagging of words). We train the self-supervising model on the preprocessed Dataset 1 and Dataset 2 corpus. After tuning the hyperparameters of the Word2Vec model, we expect the Word2Vec model to have relatively good predictive accuracy uses a sliding window of 8 words to determine the context of the neighbouring words considered for predicting the target word. Since the training corpus is relatively small, we use a 0.005 learning rate, and minimum word count of 1.

While we do not expect this Word2Vec model to perform robust embedding predictions since

it is trained on a smaller corpus, we hope to shrink the gap between the LSTM's inputted embeddings and its output on the predictions of Dataset 2 sentiments, to improve on the model's accuracy. By computing embeddings on the text specific to our target task, we will at least know that our word embeddings are meaningful for our target dataset. Therefore, we iterate the word2vec training over 50 epochs, to improve its training accuracy.

Model 11: After training the Word2Vec model on Datasets 1 and 2, input the embedded Dataset 1 into the LSTM Model 6.

Step 3: Identifying Correlation of BoC tone with changes in Asset Prices

Building upon predictions from Step 2, we aim to observe the relation between the polarity of central bank announcements and changes in asset prices. After aggregating the paragraph sentiment labels to produce an overall sentiment for each article via majority voting, we plan to employ an event study analysis to investigate this relation. Event study analysis is a widely used methodology for analyzing the effects of specific events on financial markets (Campbell et al., 1997). It allows us to isolate the influence of central bank announcements by focusing on a defined event window surrounding the announcement date. Due to the difficulty in accessing free high-frequency timestamped data, we limited our plan to work with daily frequency data (Dataset 3) for the stock prices.

The event window is defined as a 3-day period: the day before the announcement, the announcement day itself, and the day after the announcement. To analyze stock returns, we calculate the average returns or price changes within the event window for each announcement date. Specifically, we determine the absolute return for each day within the event window by comparing the stock price on that day to the previous day's price. The cumulative return for the event window is obtained by summing the absolute returns.

To establish a baseline for comparison, we identify non-event periods of similar length to the event window. Cumulative returns are calculated for these non-event periods using the same methodology as for the event windows.

To assess the statistical significance of the observed cumulative returns within the event windows, we conduct a statistical test such as the Mann-Whitney U test. This allows us to compare the cumulative returns within each event window to the cumulative returns for the rest of the observation period. By evaluating the test results, we can determine whether the cumulative returns within the event windows are significantly different from those during non-event periods.

Results and Conclusions

a) Using Pretrained Word2Vec Embeddings

Model	Predictive Accuracy of LSTM Model on Dataset 2
Model 6	0.480
Model 7	0.496
Model 8	0.504
Model 9	0.490
Model 10	0.449

b) Using Word2Vec Embeddings trained on Datasets 1 and 2

Model	Predictive Accuracy of LSTM Model on Dataset 2
Model 11	0.472

With the Word2Vec CBOW embedding model trained on Datasets 1 and 2, the LSTM model's predictive accuracy is similar with the accuracy seen from the LSTM models using the pre-trained Word2Vec embeddings. This result suggests that the low predictive accuracy of our model on the BoC sentiments are not due to poorly embedded inputs.

Compared to the models with trainable embeddings (Model 1-5), the pre-trained Word2Vec embedding inputs suggest an improvement of ~10% test accuracy (table a). However, this test accuracy remains too low to yield robust predictions. We suspect the following reasons:

1. Dataset 1's sample size is too small. This is obvious when the test accuracy increases from Model 1-5 to Model 6-9, where the data splitting format changed from training-test-validation to training-test or only training.

2. The discrepancy in text length between Dataset 1 (short sentences) and Boc announcement paragraphs (containing 5-6 short sentences). The model may struggle to accurately process and understand longer passages of text when it has primarily been trained on shorter sentences. The difference in text structure and length may pose a challenge for the model to capture the contextual dependencies and nuances present in the test data.

We propose the following potential solutions:

- Search for a larger dataset with more representative characteristics
- Manually label more BoC announcement paragraphs to create our own training data
- Switch to analyze announcement titles, which has similar length to the phrasebank sentences

The low predictive accuracy could affect the reliability of the subsequent stock return analysis. As a result, we are unable to proceed with **Step 3** as initially intended. Given more time and resources, we are confident about addressing the limitations, refining our methodology, and uncovering meaningful discoveries in this area of research.

References

1. Hayo, Bernd, and Matthias Neuenkirch. "Domestic or US news: what drives Canadian financial markets?." *Economic Inquiry* 50.3 (2012): 690-706.

2. Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.

3. Rosa, Carlo, and Giovanni Verga. "The impact of central bank announcements on asset prices in real time." *Thirteenth issue (June 2008) of the International Journal of Central Banking*(2018).

4. Schmeling, Maik, and Christian Wagner. "Does central bank tone move asset prices?." *Available at SSRN 2629978* (2019).

5. Smith, Fergal. “TSX ends lower as Bank of Canada rate hike spooks investors.” *Reuters*, June 7, 2023.

6. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

7. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Chapter 11, Deep Learning. MIT Press.

8. Mikolov, T., et al. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

9. Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (1997). The Econometrics of Financial Markets. Princeton University Press.

10. Malo, P., Sinha, A., Takala, P., Korhonen, P. and Wallenius, J. (2013): “Good debt or bad debt: Detecting semantic orientations in economic texts.” Journal of the American Society for Information Science and Technology. (in Press)]

Appendix

1. BoC Web Scraping source code, written by our term paper team members with help from TA Chenyue Liu. https://drive.google.com/file/d/1WMozwWoGq97ORIDqUpYTnmfA4dNkS01b/view?usp=drive_link

2. Data preprocessing code (code blocks 3 - 6). https://drive.google.com/file/d/1VkvT2g1Ag7_cwDuvtyl4DJMdC1lrGjVN/view?usp=drive_link

3. LSTM model 1-5 code (code blocks 21 - 29). https://drive.google.com/file/d/1VkvT2g1Ag7_cwDuvtyl4DJMdC1lrGjVN/view?usp=drive_link

4. LSTM model 6-10 code. https://drive.google.com/file/d/1QUA1wGMzCb98b582M2zWxI1mYYX-0bFA/view?usp=drive_link

5. LSTM model 11 code. https://drive.google.com/file/d/1qR2KvYvwehNdvtFKy-dtMmXDGHDk7m5N/view?usp=drive_link