Final Project Report on "The Voice of Monetary Policy"

MGT6769 – Fixed Income Securities

Submitted by:

Akinhanmi, Samuel

Aparicio, Valentina

Hales, Morgan

Kittisut, Nattakorn

Sarkari, Zenobia

Tuahivaatetonohiti, Guillaume

Date:

04/26/2023

Georgia Tech

Table of Contents

1. Introduction

1.1. Replicated Paper Summary

Our research paper of interest is called "The Voice of Monetary Policy" by Yuriy Gorodnichenko, Tho Pham, and Oleksandr Talavera issued in March 2021. This paper departs from a famous idea, called the 7-55-38 rule of communication, which explains that words deliver only 7 percent of a message, body language accounts for 55 percent, and voice tone explains the remaining 38 percent. As a result, it seems clear to these authors that effective communication must rely on more than just plain words. The objective of this paper now becomes to examine the influence of body language and voice tone on financial markets, such as the stock market, bond market, amongst others. To do so, the authors will develop a deep learning model with the goal of detecting emotions embedded in press conferences after the meetings of the Federal Open Market Committee (FOMC). This model is called a neural network, which is a machine-learning algorithm which can be trained and customized to recognize emotions from voice variations. Using this model, and after experimenting with different hyper-parameters, the authors managed to design an 85 percent accurate model. Throughout the results section, the authors finally find that, after controlling for the Federal Reserve actions and the sentiment in policy texts, positive voice tone in the voices of Fed Chairs – such as Jerome Powell, Ben Bernanke, and Janet Yellen – leads to statistically significant and economically large increases in share prices. Thus, voice tone does seem to have an effect on the stock market. However, the bond market appears to be indifferent to voice tone. Consequently, the authors conclude that their results provide sound reasons for improving the effectiveness of central bank communications.

1.2. Our Approach

To ensure that our research findings could be replicated, we took great care to use the same training data as the original study. Additionally, we recorded the Q&A sessions of Jerome Powell during his testimony before Congress on March 8th, 2023, and the FOMC on March 22nd, 2023, using Audacity.
Our approach involved utilizing SkLearn to implement a simple Recurrent Neural Network (RNN) initially. However, since our research aim was to extract emotional sentiment scores, we subsequently customized the classifier to achieve this objective. In doing so, we relied on a range of advanced techniques and methodologies, including deep learning algorithms, natural language processing (NLP) techniques, and sentiment analysis models. Our modifications allowed us to extract nuanced emotional sentiment scores from Jerome Powell's responses during both Q&A sessions.
By leveraging these techniques, we were able to develop a more nuanced understanding of the emotional sentiment underlying Jerome Powell's responses. In turn, this provided valuable insights into the broader economic landscape and helped us to make more accurate predictions about future market trends.

2. Data Preparation

2.1. Training Data

To train our system, we utilized primarily two datasets: RAVDESS and TESS. Both datasets contain pre-recorded audio samples of actors reading the same sentences with different emotional expressions.

We relied on these datasets because we lacked information about Jerome Powell's emotional states during his speeches. By leveraging pre-recorded samples, we were able to create a training dataset that encompassed a wide range of emotional expressions, enabling our system to recognize and respond to different emotional cues in Jerome Powell's speeches.

While we acknowledge that using pre-recorded samples is not ideal, it allowed us to develop a more robust and effective model that could better respond to the nuances of human speech. Overall, our approach highlights the importance of leveraging high-quality training data to develop more effective and accurate models for analyzing complex speech data.

2.2. Testing Data

To test the effectiveness of our system, we utilized Audacity to record the Q&A sections of Federal Reserve videos posted on YouTube. Specifically, we recorded approximately 700 Q&A responses to Jerome Powell's speeches, with a focus on the testimony before Congress and the FOMC meetings.

To ensure the validity of our data, we employed a randomized cross-validation process. This allowed us to evaluate the performance of our system across a range of different test samples, helping us to identify and address any potential biases or anomalies in our dataset.

3. Methodology

3.1. Model

In our pursuit of creating a model capable of accurately identifying the emotions present in audio recordings, we opted to use a Recurrent Neural Network (RNN) - a type of deep learning model that enables previous outputs to be used as inputs while also containing hidden layers. Neural networks in general aim to teach computers how to process data in a way that is inspired by the human brain, and have become widely utilized for various tasks, such as music generation, sentiment classification, and machine translation. In our implementation, we leveraged the RNN model as a sentiment classifier that takes audio measurements as inputs and classifies them into one of five types of emotions: happy, positively surprised, neutral, sad, and angry. This will help us to identify the emotional undertones present in speeches given by Jerome Powell, providing valuable insights into his overall emotional state and helping us to make more informed predictions about future market trends.

While there are several benefits to utilizing RNNs, such as the ability to process input of any length and having an internal memory that remembers previous inputs, there are also several drawbacks. These include slow computation and the potential for vanishing or exploding gradients, which can complicate the training process. However, if applied properly, RNNs are a powerful algorithm capable of producing confident predictions.

4. Results

The sentiment analysis program utilized in this study generates a score between -1 and 1, with a score of 0 indicating a neutral sentiment. Scores between -1 and 0 indicate a negative sentiment aligned with emotions such as anger or disgust, while scores between 0 and 1 indicate a positive sentiment associated with emotions like happiness or surprise. These sentiment scores from the testing set were found to have a measurable impact on market returns, bond yields, and currency strength. To expand the scope of the study, the research team analyzed Jerome Powell's testimony at the semiannual monetary policy report hearing in Congress, consisting of 692 answers, in order to obtain the sentiment scores for each response. This additional data set allowed for a more robust analysis of the relationship between sentiment and market outcomes.

```
[24]: feature=extract_feature("C:\\Users\\guill\\D
      feature=feature.reshape(1,-1)

      prediction=loaded_model.predict(feature)
      prediction

[24]: array(['disgust'], dtype='<U7')

[25]: score

[25]: -0.694
```

Figure 1: Disgust Example Code Snippet

Figure 1 displays a question posed to Jerome Powell during his testimony, "I'm not trying to trick you, you are raising interest rate […] to slow the economy, are you not?" Powell responded with a tone of disgust, "Yes, to cool the economy off".

The sentiment analysis tool used in the study recorded a negative sentiment score of -0.694 for this response, indicating a strong negative emotion in Powell's tone.

A similar scenario occurred when he was asked a follow-up question about the impact of slowing the economy on employment. In response, Powell said, "Not really, we're trying to restore price stability, not wages", with a sentiment score of -0.879, indicating an 'angry' emotion.

Furthermore, the analysis showed a noteworthy 3.43% decrease in the S&P 500 Index following Powell's press conference. Conversely, when the data from the Federal Open Market Committee was analyzed, the results showed fewer negative responses, resulting in an overall increase of 5.19% in the market. However, it is important to note that correlation does not imply causation.

Figure 2: S&P 500 Movements

It should be noted that Powell was subjected to a barrage of questions for over three hours, which led to his comments becoming more hawkish and causing a downturn in market sentiment. Additionally, it is important to consider that several crucial economic data points were also released later that same week, such as the employment report.

Another metric used to measure the market's sentiment is the VIX, which is an indicator of implied volatility and market fear. A higher VIX indicates more fear in the market. Following Jerome

Powell's press conference, which had predominantly negative emotions, the VIX rose significantly, indicating an increase in market fear. On the other hand, the FOMC meeting produced fewer negative emotions, resulting in a slight increase in the VIX.

Cryptocurrencies, despite their supposed decentralization, still shows sensitivity to global economic news and events. Although the decentralized nature is often quoted as a key advantage over FIAT currencies, market sentiment can still be influenced by several external factors, in this case economic policy decisions. Although the crypto market is expecting a decoupling from other markets, at this moment to remain competitive it is essential to remain informed about Global economic trends and events.

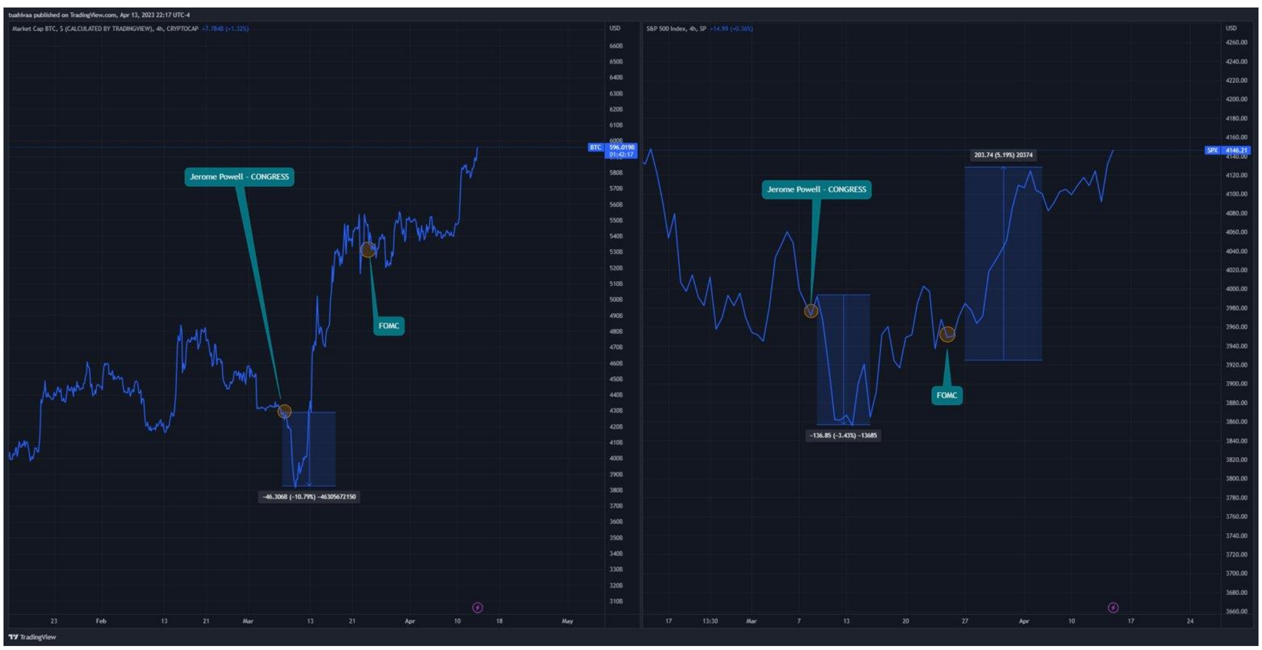

Figure 3: Bitcoin (BTC) and S&) 500 Movement Comparison

5. Conclusion

Our replication study strongly suggests that the market closely monitors Chairman Jerome Powell's responses to questions, and his tone of voice is just as critical. There are indications that his access to confidential information could manifest through his voice intonation. This field of study is still in its infancy, and as more recordings of his voice become available, the results can only improve. However, a significant limitation of our study lies in the selection of questions. While it is feasible to predict emotions from answers, some questions are more significant than others. Additionally, the way a question is phrased is crucial. If a negative emotion is detected, is it because Powell does not want to answer, or is the question poorly worded, indicating a specific agenda or narrative? Although this study focused solely on analyzing Jerome Powell's tone of voice, it is important to note that his body language and micro-expressions can also provide valuable information about his emotional state during press conferences. While these aspects were not included in the current study, they can be considered in future research to add even more insight into Powell's communication style. By analyzing multiple forms of nonverbal communication, a more comprehensive understanding of Powell's emotional state and the potential impact on the market could be achieved.

6. Appendix

Python Code

```
import os
Root = [DIRECTORY]
os.chdir(Root)
import librosa
import soundfile
import os, glob, pickle
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
# Emotions in the RAVDESS dataset
emotions={
  '01':'neutral',
  '02':'calm',
  '03':'happy',
```

```
  '04':'sad',
  '05':'angry',
  '06':'fearful',
  '07':'disgust',
  '08':'surprised'}
#Emotions to observe – change at will
observed_emotions=['calm', 'happy', angry, 'disgust']
#Load the data and extract features for each sound file
def load_data(test_size=0.2):
   x,y=[],[]
   for file in glob.glob("C:\\Users\\guill\\Downloads\\speech-emotion-recognition-ravdess-data\\Actor_*/*.wav"):
      file_name=os.path.basename(file)
      emotion=emotions[file_name.split("-")[2]]
      if emotion not in observed_emotions:
         continue
      feature=extract_feature(file, mfcc=True, chroma=True, mel=True)
      x.append(feature)
      y.append(emotion)
   return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
#Extract features (mfcc, chroma, mel) from a sound file
def extract_feature(file_name, mfcc, chroma, mel):
   with soundfile.SoundFile(file_name) as sound_file:
      X = sound_file.read(dtype="float32")
      sample_rate=sound_file.samplerate
      if chroma:
         stft=np.abs(librosa.stft(X))
      result=np.array([])
      if mfcc:
         mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
         result=np.hstack((result, mfccs))
      if chroma:
         chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
         result=np.hstack((result, chroma))
      if mel:
         mel=np.mean(librosa.feature.melspectrogram(S=stft).T,axis=0)
         result=np.hstack((result, mel))
   return result
#Split the dataset
x_train,x_test,y_train,y_test=load_data(test_size=0.25)
#Get the shape of the training and testing datasets
print((x_train.shape[0], x_test.shape[0]))
#Get the number of features extracted
print(f'Features extracted: {x_train.shape[1]}')
#Initialize the Multi Layer Perceptron Classifier
model=MLPClassifier(alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,), learning_rate='adaptive', 
max_iter=500)
#Train the model
model.fit(x_train,y_train)
#Predict for the test set
y_pred=model.predict(x_test)
y_pred
```

```
#Calculate the accuracy of our model
accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)
#Print the accuracy
print("Accuracy: {:.2f}%".format(accuracy*100))
from sklearn.metrics import accuracy_score, f1_score
f1_score(y_test, y_pred,average=None)
import pandas as pd
df=pd.DataFrame({'Actual': y_test, 'Predicted':y_pred})
df.head(20)
import pickle
# Writing different model files to file
with open( 'modelForPrediction1.sav', 'wb') as f:
    pickle.dump(model,f)
filename = 'modelForPrediction1.sav'
loaded_model = pickle.load(open(filename, 'rb')) # loading the model file from the storage
feature=extract_feature("DIRECTORY", mfcc=True, chroma=True, mel=True)
feature=feature.reshape(1,-1)
prediction=loaded_model.predict(feature)
prediction
feature=extract_feature("DIRECTORY", mfcc=True, chroma=True, mel=True)
feature=feature.reshape(1,-1)
prediction=loaded_model.predict(feature)
prediction
```