Assignment 10: Fine-Tuning PLM for Monetary Policy Stance Classification

Name: Shiqi Hu

GTID: 904061372

Step 0: Background Research

Define Hawkish and Dovish

- **Hawkish**: In monetary policy, it refers to a stance where the Federal Reserve favors raising interest rates and reducing the money supply. This approach often includes quantitative tightening (selling Treasury securities to tighten liquidity), aiming to curb inflation, but it can also slow economic growth.
- **Dovish**: In contrast, dovish refers to a stance where the Federal Reserve favors lower interest rates and an increase in money supply. The Fed may pursue quantitative easing by buying Treasuries, which stimulates economic growth but can lead to higher inflation if sustained.

Key Monetary Policy Events

- **Dot-com Bubble Burst (2000-2001)**: Following rapid economic growth and market speculation, the Fed raised interest rates multiple times in the late 1990s. When the bubble burst in 2000, to stimulate the economy, the Fed quickly cut rates from 6.51% in Nov 2000 to 1.82% in Dec 2001, a shift that marked a significant response to tech sector volatility.
- **2008 Financial Crisis**: In response to the collapse of major financial institutions, the Fed slashed interest rates from 5.26% in July 2007 to near zero in Dec 2008 and introduced unprecedented quantitative easing (QE) measures, purchasing large amounts of Treasuries and mortgage-backed securities to support financial stability and revive the economy.
- **2015-2018 Rate Hikes**: After years of near-zero interest rates post-2008, the Fed gradually raised rates starting in 2015, reflecting a more hawkish stance as the economy recovered. This marked a shift toward normalization of monetary policy following the crisis-era support.
- **COVID-19 Pandemic (2020)**: Facing a severe economic downturn, the Fed once again cut interest rates to near zero and launched aggressive QE, buying massive amounts of Treasuries and other securities. This dovish approach was aimed at supporting liquidity and economic recovery amid widespread disruptions.
- **2022-2023 Inflation Surge and Rate Hikes**: In response to historic inflation, the Fed rapidly increased interest rates starting from near zero in March 2022 to around 5.33% in Aug 2023, implementing the most aggressive rate hikes in decades. This hawkish policy was designed to curb inflation but raised concerns about potential recessionary impacts.

Step 1: Fine-Tuning RoBERTa

Q: Why tokenization is important for models like RoBERTa?

- Tokenization is essential for models like RoBERTa because it breaks down text into smaller pieces ("tokens") that the model can understand and process.

- Specifically, RoBERTa uses a specific tokenizer to convert words into numerical representations, which the model requires for learning patterns in the text.
- Setting padding='max_length' ensures that all inputs have the same length by adding padding where necessary, while truncation=True cuts off longer texts to a maximum length of 256 tokens. This standardization helps RoBERTa process batches of text efficiently and avoids errors from variable-length inputs.

Q: What does the confusion matrix tell you about the model's performance on hawkish, dovish, and neutral labels?

- After conducting hyperparameter tuning with the training and validation sets (where 10% of the full training data was used for validation), the model achieved an F1-score of 0.70 on the test set. This indicates that the model's classification performance is fairly balanced across labels.

- The confusion matrix shows the following insights about the model's performance:

  - **Dovish Labels**: The model correctly classified 49 out of 69 "Dovish" instances, with a few misclassifications primarily into the "Neutral" category (16 instances), indicating some confusion between dovish and neutral tones.
  - **Hawkish Labels**: The model performs best on "Hawkish" instances, correctly classifying 41 out of 49, with minimal confusion. This suggests that the model has a stronger ability to identify hawkish language.
  - **Neutral Labels**: The model correctly identified 58 out of 96 "Neutral" cases but misclassified 21 as "Dovish" and 17 as "Hawkish." This pattern indicates that neutral language can be challenging to distinguish, especially from dovish tones.

Q: How does performance change as you vary learning rate, batch size, and number of epochs?

My parameters include:

- learning_rates = [2e-5, 5e-5, 1e-4]
- batch_sizes = [4, 8, 16]
- num_epochs = [3, 5, 10]

Performance analysis on the validation set based on hyperparameters (F1-score):

- Impact of Learning Rate on F1 Score:

  - **Learning Rate = 2.00e-05**: Generally performs well with batch sizes 4, 8, and 16, especially as the number of epochs increases. F1 scores range from around 0.538 to 0.678, with higher epochs and larger batch sizes (e.g., batch size 16, epochs 5 or 10) showing improvements in performance.
  - **Learning Rate = 5.00e-05**: Shows the best overall F1 scores, especially with batch sizes of 8 and 16 and higher epochs. The highest F1 scores (around 0.723 and 0.720) are achieved with a batch size of 16 and epochs of 5 and 10, respectively. This learning rate provides a good balance between stability and performance, as it consistently outperforms the others.
  - **Learning Rate = 1.00e-04**: F1 scores remain very low (around 0.192) across all configurations of batch size and epochs, suggesting this learning rate is too high and may lead to unstable or underperforming models.

- Impact of Batch Size on F1 Score:

  - **Batch Size = 4**: With lower batch sizes, the model performance is less consistent. For learning rates of 2.00e-05 and 5.00e-05, F1 scores are generally lower compared to larger batch sizes, even when the number of epochs increases. The highest F1 score for this batch size is around 0.678 with a learning rate of 2.00e-05 and 10 epochs, indicating that increasing epochs helps compensate for the smaller batch size.
  - **Batch Size = 8**: Batch size 8 provides better performance stability than batch size 4, particularly with learning rates of 2.00e-05 and 5.00e-05. The F1 score reaches around 0.646 and 0.602 with 5 and 10 epochs respectively for 2.00e-05, and around 0.670 with 5.00e-05 and 10 epochs, indicating improved performance with higher epochs.
  - **Batch Size = 16**: The best-performing models are found with batch size 16, especially with a learning rate of 5.00e-05. The highest F1 scores of 0.723 and 0.720 are achieved with epochs of 5 and 10 for the 5.00e-05 learning rate, indicating that larger batch sizes are beneficial for this setup. However, with a learning rate of 1.00e-04, performance remains poor across all epochs, suggesting that an optimal learning rate is necessary to benefit from larger batch sizes.

- Number of Epochs:

  - **3 Epochs**: Models with fewer epochs tend to underperform, especially with higher batch sizes. The F1 score for 3 epochs maxes out at around 0.670 for batch size 16 and a learning rate of 5.00e-05, indicating that higher epochs are generally more effective for this problem.
  - **5 Epochs**: Increasing the number of epochs to 5 typically improves performance, especially with a batch size of 16 and a learning rate of 5.00e-05, achieving the highest F1 score of 0.723. For lower learning rates and smaller batch sizes, the improvements are moderate but noticeable.
  - **10 Epochs**: Further increasing epochs to 10 shows mixed results, with some combinations (e.g., 5.00e-05, batch size 16) maintaining high F1 scores around 0.720. However, additional epochs seem to provide diminishing returns, especially for smaller batch sizes and suboptimal learning rates.

Summary of Findings

- Optimal Combination: A learning rate of 5.00e-05, batch size of 16, and 5 epochs yield the best F1 scores (around 0.723).
- Learning Rate Sensitivity: The model is highly sensitive to the learning rate, with 2.00e-05 and 5.00e-05 performing well, while 1.00e-04 results in consistently low F1 scores.
- Batch Size and Epoch Balance: Larger batch sizes (16) combined with moderate epochs (5 to 10) provide the best performance, while smaller batch sizes benefit from higher epochs.

	learning_rate	batch_size	epochs	f1_score	accuracy	model_path
0	2.00e-05	4	3	0.358995	0.453488	./hsq/results_lr2e-05_bs4_epochs3
1	2.00e-05	4	5	0.670833	0.674419	./hsq/results_lr2e-05_bs4_epochs5
2	2.00e-05	4	10	0.678992	0.686047	./hsq/results_lr2e-05_bs4_epochs10
3	2.00e-05	8	3	0.583810	0.593023	./hsq/results_lr2e-05_bs8_epochs3
4	2.00e-05	8	5	0.646587	0.651163	./hsq/results_lr2e-05_bs8_epochs5
5	2.00e-05	8	10	0.603134	0.616279	./hsq/results_lr2e-05_bs8_epochs10
6	2.00e-05	16	3	0.664012	0.674419	./hsq/results_lr2e-05_bs16_epochs3

	learning_rate	batch_size	epochs	f1_score	accuracy	model_path
7	2.00e-05	16	5	0.643805	0.651163	./hsq/results_lr2e-05_bs16_epochs5
8	2.00e-05	16	10	0.638975	0.639535	./hsq/results_lr2e-05_bs16_epochs10
9	5.00e-05	4	3	0.556324	0.569767	./hsq/results_lr5e-05_bs4_epochs3
10	5.00e-05	4	5	0.647730	0.651163	./hsq/results_lr5e-05_bs4_epochs5
11	5.00e-05	4	10	0.610584	0.627907	./hsq/results_lr5e-05_bs4_epochs10
12	5.00e-05	8	3	0.549919	0.569767	./hsq/results_lr5e-05_bs8_epochs3
13	5.00e-05	8	5	0.532493	0.593023	./hsq/results_lr5e-05_bs8_epochs5
14	5.00e-05	8	10	0.602707	0.604651	./hsq/results_lr5e-05_bs8_epochs10
15	5.00e-05	16	3	0.670251	0.674419	./hsq/results_lr5e-05_bs16_epochs3
16	5.00e-05	16	5	0.723151	0.732558	./hsq/results_lr5e-05_bs16_epochs5
17	5.00e-05	16	10	0.720359	0.720930	./hsq/results_lr5e-05_bs16_epochs10
18	1.00e-04	4	3	0.192837	0.406977	./hsq/results_lr0.0001_bs4_epochs3
19	1.00e-04	4	5	0.192837	0.406977	./hsq/results_lr0.0001_bs4_epochs5
20	1.00e-04	4	10	0.192837	0.406977	./hsq/results_lr0.0001_bs4_epochs10
21	1.00e-04	8	3	0.192837	0.406977	./hsq/results_lr0.0001_bs8_epochs3
22	1.00e-04	8	5	0.192837	0.406977	./hsq/results_lr0.0001_bs8_epochs5
23	1.00e-04	8	10	0.192837	0.406977	./hsq/results_lr0.0001_bs8_epochs10
24	1.00e-04	16	3	0.192837	0.406977	./hsq/results_lr0.0001_bs16_epochs3
25	1.00e-04	16	5	0.192837	0.406977	./hsq/results_lr0.0001_bs16_epochs5
26	1.00e-04	16	10	0.192837	0.406977	./hsq/results_lr0.0001_bs16_epochs10

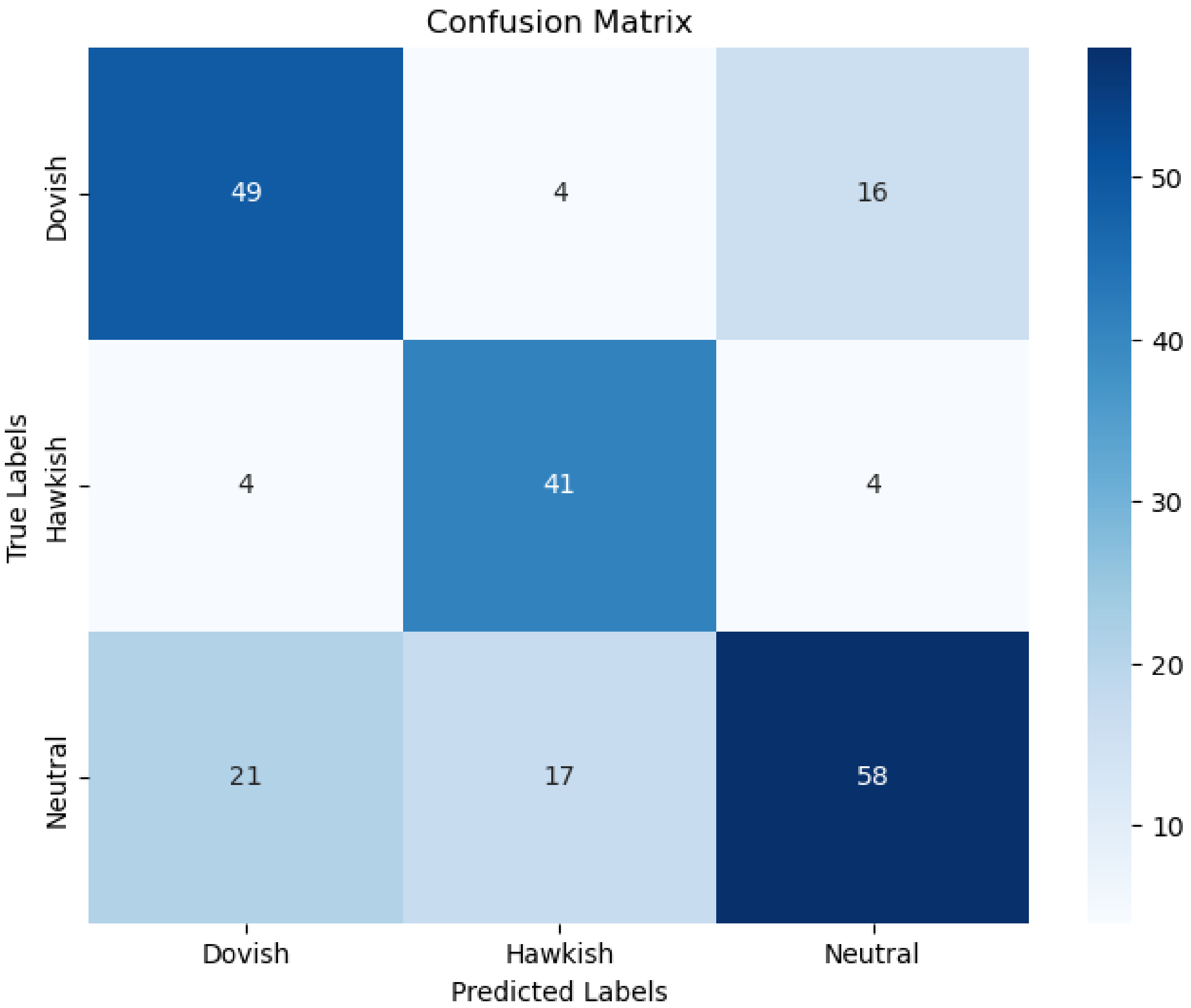
```
Best Combination: Learning Rate=5e-05, Batch Size=16, Epochs=5




Classification Report on Test Dataset:
              precision    recall  f1-score   support

      Dovish       0.66      0.71      0.69        69
     Hawkish       0.66      0.84      0.74        49
     Neutral       0.74      0.60      0.67        96

    accuracy                           0.69       214
   macro avg       0.69      0.72      0.70       214
weighted avg       0.70      0.69      0.69       214
```



Step 2: Inference Using the Fine-Tuned Model

1. **Filtering and Inference on FOMC Meeting Minutes:**

- Tokenize the raw text files into sentences.
- Before running inference, filter the sentences from the FOMC meeting minutes (January 1996 to September 2024) based on the presence of words from **Panel A1** or **B1** in Table 1 of the paper. Only sentences containing these keywords should be kept for further analysis.
- Once the sentences are filtered, use the fine-tuned model to perform inference on the remaining sentences. Mark each sentence as hawkish, dovish, or neutral.
- Save the predictions to a file, specifying whether each sentence is hawkish, dovish, or neutral.

2. **Construct a Hawkishness Measure:**

- Following the method described in the paper, construct a **document-level hawkishness measure**. For each document (FOMC meeting minute), compute:

$$\text{Hawkishness} = \frac{# \text{Hawkish sentences} - # \text{Dovish sentences}}{# \text{Total filtered sentences}}$$

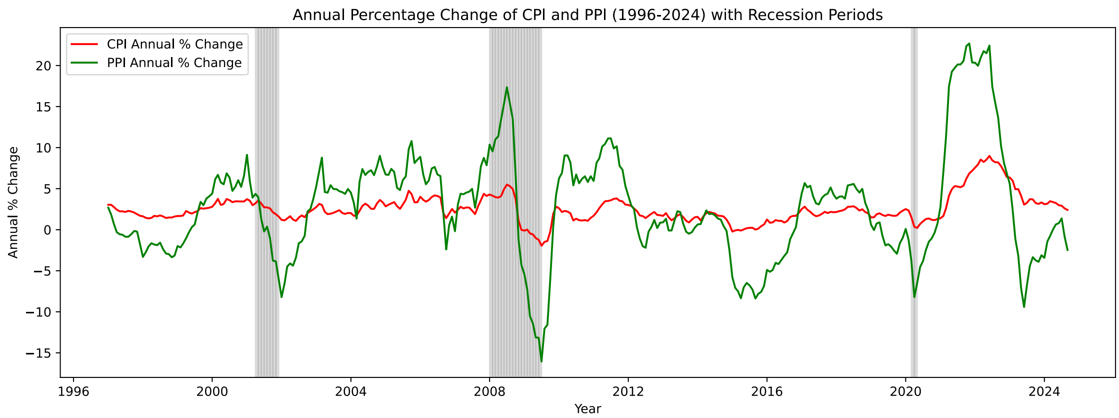
- Save these results for the next step.

Step 3: Analysis of CPI, PPI, and Recession

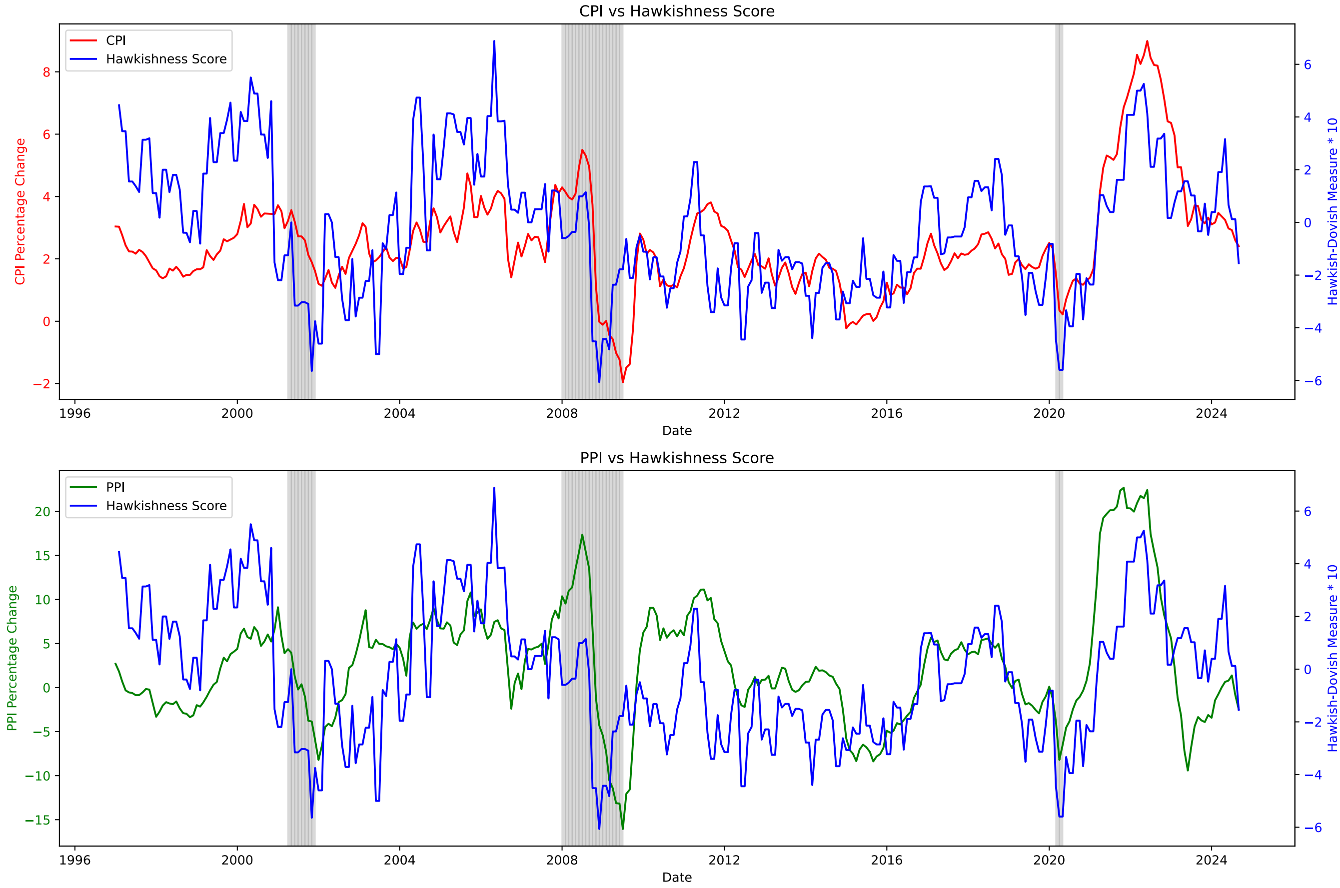
Provide commentary on what you observe in the relationship between monetary policy stance (hawkish/dovish), CPI, PPI, and periods of recession. Do you see any patterns in how the hawkishness measure relates to inflation or economic downturns?

- **Hawkishness and Inflation Relationship**:
    - The Fed typically becomes more hawkish as inflation rises - we see the hawkishness measure increasing alongside climbing CPI and PPI, as demonstrated in both the early 2000s and 2021-2022 inflationary periods. Conversely, when inflation metrics decline, the Fed adopts a more dovish stance by lowering interest rates and expanding money supply to support economic growth.
- **Economic Downturns and Dovish Shifts**:
    - During recessions, there's a clear pattern of the hawkishness measure dropping sharply as the Fed pivots to dovish policies - lowering rates and increasing market liquidity to stimulate economic recovery. This was particularly evident during the 2008 financial crisis and 2020 pandemic recession.
- **Lagged Effects and Policy Impact**:
    - There appears to be a slight lag between peaks in the hawkishness measure and the subsequent stabilization or decline in inflation (both CPI and PPI). This suggests that the effects of restrictive policy measures on inflation are not immediate and may take several months or quarters to materialize.

Plot - Annual percentage change of CPI and PPI over time (1996-2024), overlayed with a recession indicator:



Plot - Overlay Hawkishness measure with CPI, PPI, and recession indicator:


CPI vs Hawkishness Score
CPI
Hawkishness Score
CPI Percentage Change
Hawkish-Dovish Measure * 10
Date
PPI vs Hawkishness Score
PPI
Hawkishness Score
PPI Percentage Change
Hawkish-Dovish Measure * 10
Date