Scale vs. Domain Knowledge: Comparing FinBERT & RoBERTa on FOMC Hawkish-Dovish Classification

Low Shi-ya Amelia | E0970351

DSA4265: Take-Home Assignment 1 | February 15, 2026

Table of Contents

1. Introduction & Problem Statement 2
2. Data Sourcing & Processing 2
  2.1 Dataset 2
  2.2 Exploratory Data Analysis 3
  2.3 LLM Labelling Analysis 3
  2.4 Data Split 4
  2.5 Preprocessing & Tokenization 5
3. Methodology & Implementation 5
  3.1 Model Architectures 5
  3.2 Training Setup 5
4. Evaluation & Results 7
  4.1 Evaluation Approach 7
  4.2 Overall Performance 7
  4.3 ROC Curves 8
  4.4 Training Curves 9
  4.5 Performance by Monetary Policy Era 9
  4.6 Error Breakdown 10
5 Discussion & Critical Analysis 10
  5.1 Why Domain Pre-Training Fails: The Register Mismatch 10
  5.2 The Limits of Classification: What "Both Wrong" Reveals 11
  5.3 What Claude's Errors Tell Us About the Task 11
  5.4 Challenges & Mitigations 12
  5.5 Future Work 12
6 Conclusion 12
7 References 13

1. Introduction & Problem Statement

The Federal Open Market Committee (FOMC) sets U.S. monetary policy eight times per year. Its communications—meeting minutes, press conferences, and speeches—are scrutinized by traders. Subtle wording shifts can move billions in Treasury yields and equities. Classifying the monetary policy stance of these sentences can prove to be a valuable NLP task.

This is framed as a three-class sentence classification:

- **Hawkish** (tighter policy),
- **Dovish** (looser policy), or
- **Neutral** (factual/balanced).

The task requires capturing the complexities of hedging, conditional phrasing, and domain-specific terminology—making it well-suited to transformer models.

In this project I've decided to compare **FinBERT** (110M params, pre-trained on corporate financial text) against **RoBERTa-large** (355M params, general-purpose), building on the Trillion Dollar Words paper by Shah et al. (2023). There, Shah previously identified RoBERTa-large as the better model on this dataset (F1 = 0.717). My contribution is replicating their results and conducting an interpretive analysis of *why financial domain pre-training fails on central bank text*. This is supported by error analysis, era-level performance breakdowns, and a zero-shot LLM baseline for deeper understanding of this dataset and the topic at hand.

2. Data Sourcing & Processing

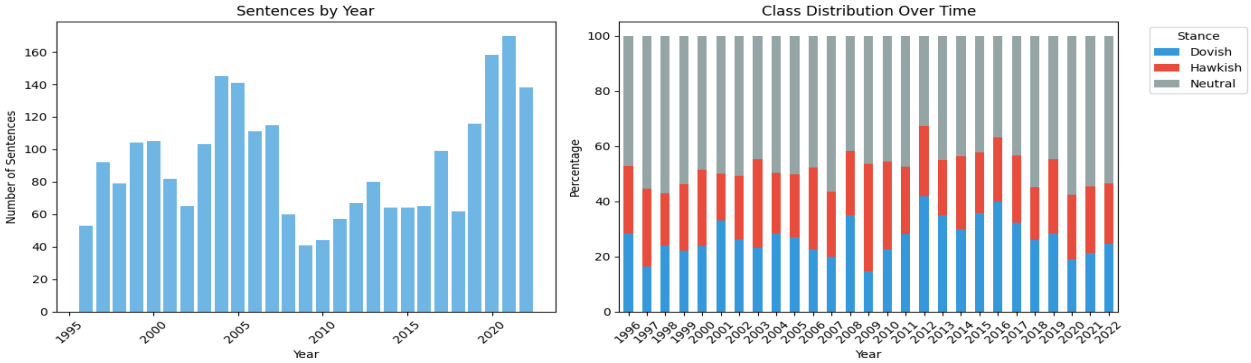
2.1 Dataset

The "Trillion Dollar Words" dataset (Shah et al., ACL 2023), available as gtfintechlab/fomc_communicationon HuggingFace, is used for this project. It consists of 2,476 expert-annotated sentences, sourced from FOMC meeting minutes, press conferences, and speeches (1996–2022), pre-split into 1,980 train / 496 test.

```
from datasets import load_dataset

dataset = load_dataset("gtfintechlab/fomc_communication")
print(f"Train size: {len(dataset['train'])}")
print(f"Test size: {len(dataset['test'])}")

df_train_raw = dataset['train'].to_pandas()
df_test_raw = dataset['test'].to_pandas()
df_all = pd.concat([df_train_raw, df_test_raw], ignore_index=True)

LABEL_MAP = {0: 'Dovish', 1: 'Hawkish', 2: 'Neutral'}
df_all['label_name'] = df_all['label'].map(LABEL_MAP)
```

2.2 Exploratory Data Analysis

The class distribution is imbalanced: ~50% Neutral, ~26% Dovish, ~24% Hawkish. This is likely because most sentences describe conditions factually rather than signal policy. Sentence lengths vary widely from 5–60+ words, with longer sentences often containing mixed signals. The dataset spans four monetary policy regimes: pre-GFC, GFC & Recovery, Normalization, and COVID & Inflation.



2.3 LLM Labelling Analysis

To further assess the annotation quality, I've built a labelling pipeline with Claude Sonnet 4.6 as my chosen LLM. This linguistic model was used as a zero-shot classifier on all 2,476 sentences:

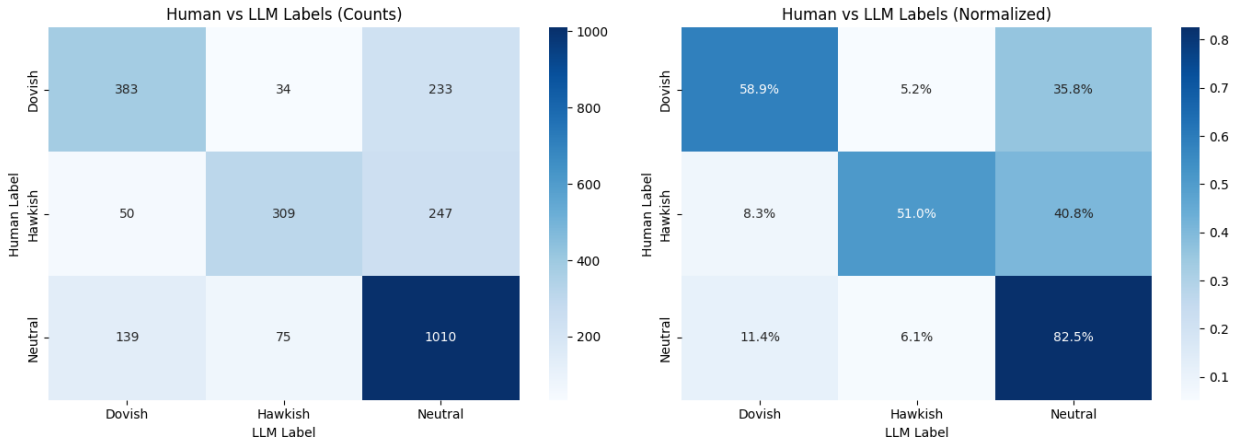
```
from anthropic import Anthropic
from google.colab import userdata

client = Anthropic(api_key=userdata.get('ANTHROPIC_API_KEY'))

SYSTEM_PROMPT = """You are a monetary policy expert classifying
sentences from Federal Reserve (FOMC) communications.
For each sentence, classify it as:
- HAWKISH: Future monetary policy tightening
- DOVISH: Future monetary policy easing
- NEUTRAL: Factual, balanced, or not clearly signaling
Consider the sentence from an investor's perspective.
Respond ONLY in format: idx: LABEL"""

def label_batch(sentences, start_idx=0):
    formatted = "\n".join(
        [f"{start_idx + i}: {s}" for i, s in enumerate(sentences)])
    response = client.messages.create(
        model="claude-sonnet-4-20250514", max_tokens=1000,
        system=SYSTEM_PROMPT,
        messages=[{"role": "user", "content":
                   f"Classify each sentence:\n\n{formatted}"}])
    # Parse "idx: LABEL" lines from response
    results = {}
    for line in response.content[0].text.strip().split("\n"):
        if ":" in line:
            parts = line.split(":", 1)
            try:
                idx = int(parts[0].strip())
                label = parts[1].strip().upper()
                if label in ['HAWKISH', 'DOVISH', 'NEUTRAL']:
                    results[idx] = label
            except (ValueError, IndexError):
                continue
    return results
```

Claude achieved 68.6% accuracy with Cohen's Kappa = 0.479 (moderate agreement). Per-class agreement: Neutral 82.5%, Dovish 58.9%, Hawkish 51.0%.

	Precision	Recall	F1	Support	Agree %
Dovish	0.670	0.589	0.627	650	58.9%
Hawkish	0.739	0.510	0.604	606	51.0%
Neutral	0.678	0.825	0.744	1224	82.5%
Macro Avg	0.696	0.641	0.658	2480	68.6%

The most common error pattern is Claude labelling Dovish/Hawkish sentences as Neutral (233 and 247 cases), suggesting the LLM defaults to conservative classification when policy signals are subtle. However, Claude rarely confuses Dovish for Hawkish (34 cases) or vice versa (50 cases). It understands the directional distinction but struggles with the signal-vs-neutral threshold. This zero-shot baseline (F1 = 0.658) sets the bar our fine-tuned models must beat.

Key examples of Claude's errors: "But I want to emphasize that we do have a commitment to raising inflation to 2 percent" (label is Dovish, Claude says Hawkish — fixating on "raising inflation" without recognising that *wanting* higher inflation implies loose policy). "The National Bureau of Economic Research determined in July that the recession had ended" (label is Hawkish, Claude says Neutral, treating this as factual citation, missing that the recession ending reduces the case for accommodation). Some of Claude's "errors" are arguably defensible: "With inflation low and resource use slack, the Committee saw no need for tightening" (label is true: Neutral, Claude says Dovish). Explicitly saying "no tightening" is a reasonable dovish reading.



2.4 Data Split

We preserved the original test set (496), but split the training data further by 85/15to form the validation set for our models. Hence the new split is as follows: 1,683 train, 297 validation, 496 test.

```
df_train_orig = df_all.iloc[:len(df_train_raw)].copy()
df_test = df_all.iloc[len(df_train_raw):].copy()

df_train, df_val = train_test_split(
    df_train_orig, test_size=0.15,
    stratify=df_train_orig['label'], random_state=42)
```

2.5 Preprocessing & Tokenization

Both models use subword tokenization with a maximum sequence length of 128. FinBERT uses WordPiece tokenization inherited from BERT, while RoBERTa uses Byte-Pair Encoding (BPE). Sentences shorter than 128 tokens are padded; longer ones are truncated.

```
from torch.utils.data import Dataset

class FOMCDataset(Dataset):
    def __init__(self, texts, labels, tokenizer, max_length=128):
        self.texts = texts
        self.labels = labels
        self.tokenizer = tokenizer
        self.max_length = max_length

    def __getitem__(self, idx):
        encoding = self.tokenizer(
            str(self.texts[idx]),
            add_special_tokens=True,
            max_length=self.max_length,
            padding='max_length',
            truncation=True,
            return_attention_mask=True,
            return_tensors='pt')
        return {
            'input_ids': encoding['input_ids'].flatten(),
            'attention_mask': encoding['attention_mask'].flatten(),
            'labels': torch.tensor(self.labels[idx], dtype=torch.long)}
```

3. Methodology & Implementation

3.1 Model Architectures

FinBERT (yiyanghkust/finbert-pretrain): BERT-base (12 layers, 110M params) further pre-trained on corporate financial text (Yang et al., 2020). I've chosen the pre-trained base, not the sentiment-fine-tuned ProsusAI variant, as this variant has already been fine-tuned further on Financial PhraseBank dataset, which might conflict with it's hawkish-dovish training.

RoBERTa-large: 24 layers, 355M params. Top performer in Shah et al. (2023). Uses dynamic masking and larger pre-training corpus than BERT.

3.2 Training Setup

Both models are trained on Google Colab using its T4 GPU, with HuggingFace's Trainer API. The models shared a train_and_evaluate pipeline, ensuring identical preprocessing, training procedures, and evaluation logic. The only differences are the hyperparameters listed in the table below. This design aims to eliminate any possibility that implementation contributes to the test results and performance is attributed purely to the models.

Hyperparameter	FinBERT	RoBERTa-large
Learning Rate	2e-5	1e-5
Batch Size / Epochs	16 / 5	8 / 5
Decay / Warmup / Precision	0.01 / 0.1 / FP16	0.01 / 0.1 / FP16
Parameters	109,754,115	355,362,819

Below are the detailed justifications on my hyperparameter allocations:

- **Learning Rate:**
    - FinBERT uses 2e-5, the standard recommended rate for fine-tuning BERT-base models (Devlin et al., 2019)
    - RoBERTa-large uses a lower 1e-5, as larger models are more sensitive to learning rate and might destabilize pre-trained weights. I've decided to half the learning rate relative to its base model, when I discovered that it is common practice.
- **Batch Size:**
    - FinBERT uses batch size 16
    - RoBERTa-large uses 8. This is primarily a GPU memory constraint — RoBERTa-large has 3.2x more parameters, so each forward/backward pass consumes roughly 3x more GPU memory on Colab's T4 (15GB VRAM).
- **Epochs (5):**
    - Both models train for 5 epochs. Since there are only 1,683 training sentences, each epoch takes relatively few gradient steps (about 105 for FinBERT, 210 for RoBERTa). This aims to ensure there are enough passes for convergence while avoiding overfitting.

  It is later shown in Section 4.4, FinBERT begins to overfit at around epoch 3, while RoBERTa stabilizes around the same point, thus proving 5 epochs as a reasonable upper bound.
- **Weight Decay (0.01):**
    - L2 regularization applied to all parameters except bias and LayerNorm weights. I've chosen to follow the default value used in the original BERT and RoBERTa papers, as they've provided the best results.
- **Warmup Ratio (0.1):**
    - The learning rate linearly increases from 0 to the target rate over the first 10% of training steps, then decays linearly to 0. This is especially necessary for RoBERTa-large, whose larger parameter space is more susceptible to early instability.
- **Max Sequence Length (128):**
    - FOMC sentences rarely exceeded 100 tokens. Setting max length to 128 captured almost all sentences, while being memory and computationally efficient.
- **Mixed Precision (FP16):**
    - Both models train in 16-bit floating point, this helped to halve the GPU memory usage and speed up computation on Colab's T4 GPU (which has dedicated FP16 Tensor Cores).
- **Optimizer (AdamW):**
    - Both models used this default optimizer in HuggingFace Trainer, which provides more consistent regularization than the original Adam optimizer.

- **Evaluation Strategy:**
    - Models were evaluated after every epoch, with macro F1 as the selection criterion. Macro F1 was chosen because it weights all three classes equally, considering the data imbalance.

```
from transformers import TrainingArguments, Trainer

training_args = TrainingArguments(
    output_dir=DRIVE_PATH + f'{output_name}_checkpoints',
    num_train_epochs=num_epochs,
    learning_rate=learning_rate,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=32,
    weight_decay=0.01, warmup_ratio=0.1,
    eval_strategy='epoch', save_strategy='epoch',
    load_best_model_at_end=True,
    metric_for_best_model='f1_macro',
    greater_is_better=True,
    logging_steps=50, seed=42, fp16=True,
    report_to='none')

trainer = Trainer(
    model=model, args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    compute_metrics=compute_metrics)
trainer.train()
```

Checkpointing note: *Colab's 15GB limit could not hold 7–8GB checkpoints produced when training RoBERTa. Because of this the final-epoch model was ultimately used, rather than the best validation checkpoint. This likely understates RoBERTa slightly, but this final model still outperforms FinBERT significantly.*

4. Evaluation & Results

4.1 Evaluation Approach

We use **macro-averaged** Precision, Recall, F1, and AUC-ROC to weight all three classes equally, ensuring the ~50% Neutral majority does not inflate the scores.

```
def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    accuracy = accuracy_score(labels, predictions)
    prec, rec, f1, _ = precision_recall_fscore_support(
        labels, predictions, average='macro')
    probs = softmax(logits, axis=1)
    labels_bin = label_binarize(labels, classes=[0, 1, 2])
    auc_score = roc_auc_score(
        labels_bin, probs, multi_class='ovr', average='macro')
    return {'accuracy': accuracy, 'precision_macro': prec,
            'recall_macro': rec, 'f1_macro': f1,
            'auc_macro': auc_score}
```

4.2 Overall Performance

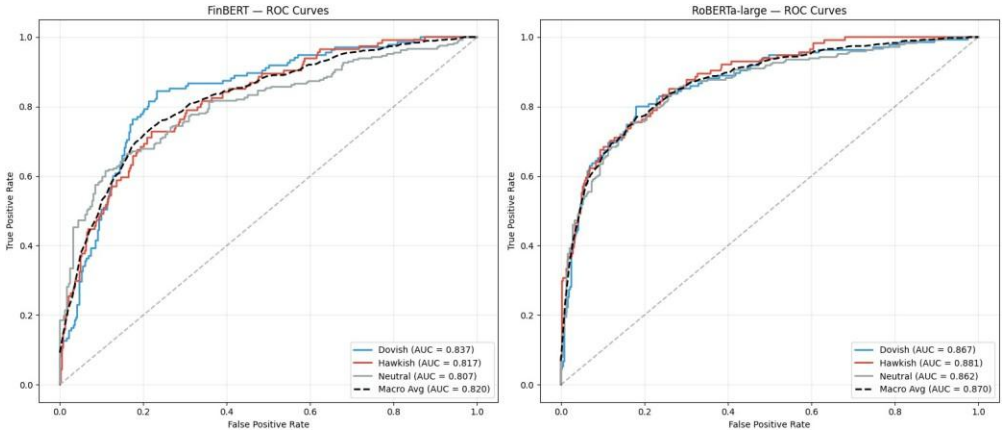
Metric	FinBERT (110M)	RoBERTa (355M)	Diff
Accuracy	0.6512	0.7258	-0.075
F1 (Macro)	0.6331	0.7102	-0.077
AUC-ROC (Macro)	0.8201	0.8698	-0.050
Training Time	128s	973s	7.6x

As shown, RoBERTa outperforms FinBERT by 5–8 points across all metrics. Also, both fine-tuned models also substantially beat Claude's zero-shot baseline (F1 = 0.658), confirming the value and relevance of task-specific fine-tuning. It is also worth noting that the RoBERTa F1 score (0.710) closely replicates Shah et al.'s (2023) reported 0.717, which nicely validates the implementation.Per-Class Breakdown

	FB P	FB R	FB F1	RB P	RB R	RB F1
Dovish	0.623	0.563	0.591	0.697	0.682	0.689
Hawkish	0.506	0.693	0.585	0.623	0.711	0.664
Neutral	0.771	0.680	0.723	0.799	0.757	0.778

Both models achieve their highest F1 on Neutral and lowest on Hawkish. FinBERT displays a notable Hawkish asymmetry: high recall (0.693) but low precision (0.506), indicating systematic over-prediction. RoBERTa maintained more balanced precision-recall trade-off across all classes.

4.3 ROC Curves



RoBERTa's per-class AUCs are tighter (0.862–0.881) while FinBERT's are more dispersed (0.807–0.837). RoBERTa's strongest class is Hawkish (0.881), FinBERT's is Dovish (0.837).

4.4 Training Curves

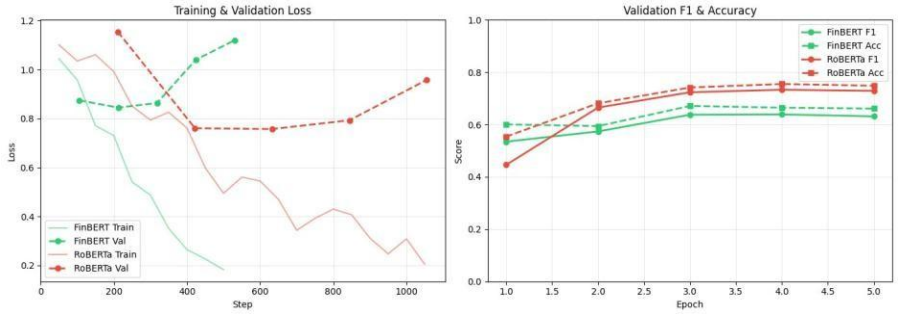
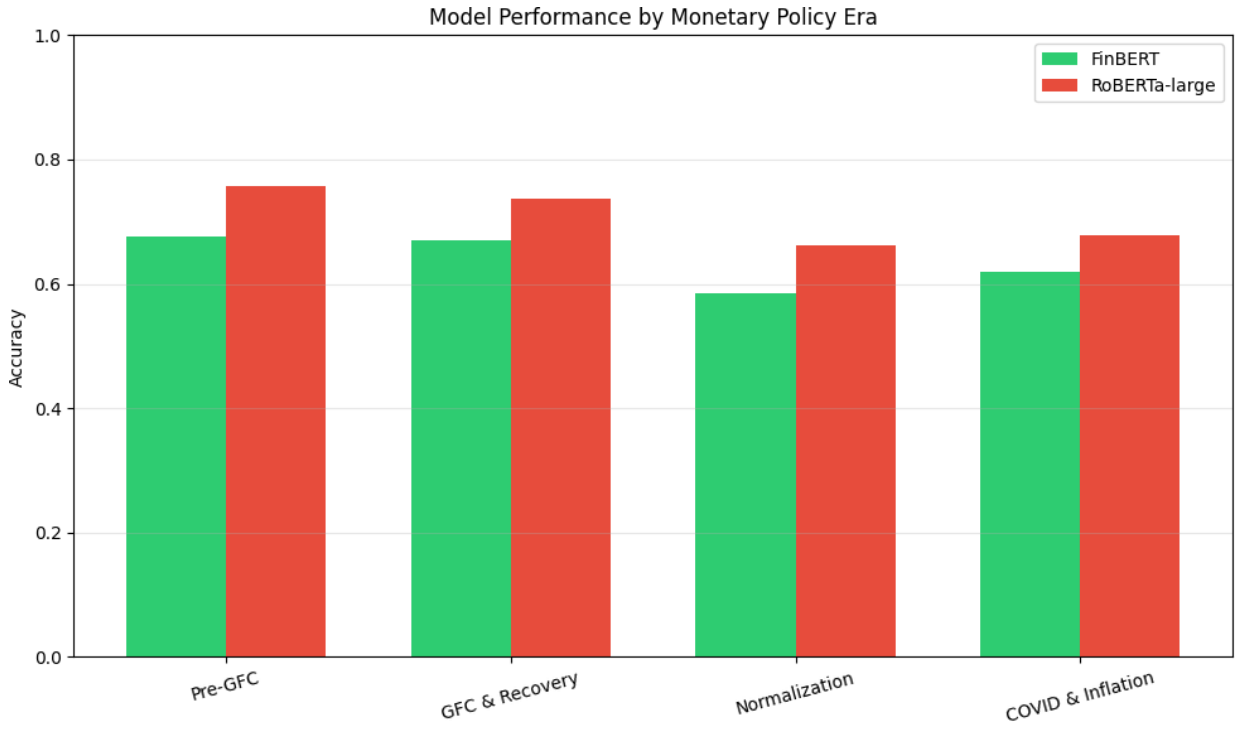


Figure 3: FinBERT overfits after epoch 3 (rising val loss); RoBERTa stabilizes by epoch 3.

Notably, FinBERT's validation loss increases after epoch 2–3 despite decreasing training loss continues dropping. This is very likely overfitting on the small training set. RoBERTa's validation loss spikes at epoch 2 before recovering slightly.

On validation F1, RoBERTa improves through epoch 3 then plateaus; FinBERT plateaus earlier and lower.

4.5 Performance by Monetary Policy Era



RoBERTa outperforms FinBERT in all four eras with a consistent 6–8 point gap. Both models perform worst during the Normalization era (2016–2019) and best during the Pre-GFC era (1996–2007).

4.6 Error Breakdown

Of 496 test sentences:

- 4.6.1 271 (54.6%) both correct
- 4.6.2 99 (20.0%) both wrong
- 4.6.3 74 (14.9%) RoBERTa-only correct
- 4.6.4 37 (7.5%) FinBERT-only correct.

FinBERT correct, RoBERTa wrong (37 cases)

The most common pattern is RoBERTa misreading neutral economic descriptions as Dovish. For example, "A few participants judged that while the labor market was close to full employment, some margins of slack remained" (2020) is Neutral, but RoBERTa predicts Dovish, apparently triggered by "slack remained." Similarly, "most members viewed a slowing to a rate closer to most estimates of the growth of the economy's potential as a favorable development" (2004) is Neutral, but RoBERTa reads "slowing" as dovish.

RoBERTa correct, FinBERT wrong (74 cases)

FinBERT's errors cluster around Hawkish over-prediction. "Participants agreed that the labor market had remained strong" (2004) is Neutral but FinBERT predicts Hawkish. More strikingly, "Very low inflation and deflation pose qualitatively similar economic problems" (2005) is Dovish — a deflation warning — but FinBERT predicts Hawkish, apparently keying on the word "inflation" regardless of context. Even the short phrase "maintaining low and stable inflation" (2001) is misclassified as Dovish when it is simply a Neutral description of the Fed's mandate.

Both wrong (99 cases)

Examples include: "Most FOMC participants anticipate that inflation will gradually move up to the 2 percent target" (2013, true: Dovish, both predict Neutral); "So we want to see that healthy process unfold" (2019, true: Dovish, both predict Hawkish); "Several commented that an asymmetric directive did not imply a commitment to tighten" (2018, true: Hawkish, both predict Neutral); "An easing of supply constraints was expected to support continued gains" (2002, true: Neutral, FinBERT says Hawkish, RoBERTa says Dovish).

5 Discussion & Critical Analysis

5.1 Why Domain Pre-Training Fails: The Register Mismatch

The consistent 7–8 point gap between RoBERTa and FinBERT across all metrics, eras, and classes demands an explanation beyond "bigger model wins." This is possibly due to a register mismatch mechanism.

FinBERT was pre-trained on corporate financial text — earnings calls, analyst reports, 10-K filings — which uses a fundamentally different linguistic register than FOMC communication. Corporate text tends to be direct ("Revenue increased 15%"). FOMC text, however, is hedged, conditional, and diplomatic ("A number of participants noted that risks were tilted to the downside"). These are different styles of communication even though both fall under the umbrella of "finance."

This mismatch explains FinBERT's specific failure patterns observed in Section 4.7. FinBERT's Hawkish over-prediction (precision 0.506) arises because corporate risk-oriented language that does more cautious assessments of economic conditions, discussions of uncertainty, overlaps with how FOMC neutral statements describe the economy. RoBERTa, having seen a broader range of registers during pre-training (news, books, web text, legal documents), is better equipped to distinguish factual economic description from policy signaling.

FinBERT's keyword-level sensitivity (Section 4.7) further supports this. Its misclassification of "Very low inflation and deflation pose similar problems" as Hawkish, suggests it has learned a strong association between the token "inflation" and hawkish sentiment from corporate text, when inflation discussions typically occur in negative/cautionary contexts. RoBERTa, with more diverse contextual exposure, processes the full phrase "very low inflation and deflation" and correctly identifies it as a dovish concern.

5.2 The Limits of Classification: What "Both Wrong" Reveals

The 99 sentences (20%) that both models misclassify define a performance ceiling for sentence-level classification on this dataset. These cases fall into several categories that illustrates the fundamental difficulty of the task. Here are some challenges they faced:

1. **Multi-step inference:** The dovish signal in "inflation will gradually move up to the 2 percent target" requires reasoning: inflation is below target => the Fed is maintaining accommodation => dovish. Neither model performs this chain; both stop at the surface level and predict Neutral. This suggests that current fine-tuning approaches learn surface-level associations rather than the causal reasoning that human annotators apply.
2. **Pragmatic implicature:** "We want to see that healthy process unfold" is dovish because of what it implies (patience, no rush to tighten) not because of what it explicitly states. The word “healthy" pulls both models toward Hawkish (strong economy means tightening), overriding the pragmatic signal of deliberate waiting. Understanding implicature remains a challenge for transformer models operating at the sentence level.
3. **Negation in policy context:** "An asymmetric directive did not imply a commitment to tighten" is Hawkish because the mere discussion of tightening signals hawkish leanings, regardless of the negation. Both models interpret the negation at face value and predict Neutral. This is a known difficulty for transformers — processing negation correctly, especially when the pragmatic meaning contradicts the literal negation.
4. **Mixed signals:** "An easing of supply constraints was expected to support gains in economic activity and a reduction in inflation" contains both positive economic signals (hawkish-adjacent) and falling inflation (dovish-adjacent). The models split: FinBERT says Hawkish, RoBERTa says Dovish. Such sentences may represent annotation ambiguity as much as model failure.

These patterns suggest that improving beyond ~75% accuracy on this dataset likely requires either document-level context (surrounding sentences disambiguate), larger training sets, or architectures designed for pragmatic reasoning.

5.3 What Claude's Errors Tell Us About the Task

Claude's zero-shot performance (F1 = 0.658) provides a unique lens on the task's difficulty. Unlike the fine-tuned models, Claude has no task-specific training — its errors reflect a general language understanding applied to domain-specific classification.

Claude's dominant failure mode — defaulting to Neutral (480 of 779 errors) — is the opposite of FinBERT's. Where FinBERT over-classifies Neutral sentences as Hawkish, Claude under-classifies Hawkish and Dovish sentences as Neutral. This reveals a threshold problem: Claude understands monetary policy concepts but sets an overly conservative bar for what constitutes a "signal" versus a "factual statement." Fine-tuning calibrates this threshold; without it, the model plays it safe.

Claude's low directional confusion (only 84 cases of Dovish↔Hawkish) is noteworthy. It rarely mistakes the direction of a signal — it just doesn't detect the signal at all. This contrasts with FinBERT, which detects signals that aren't there (Hawkish over-prediction). The two failure modes are complementary, suggesting different underlying mechanisms: FinBERT has miscalibrated domain associations, while Claude has an appropriately calibrated but overly conservative signal detector.

Interestingly, some of Claude's "errors" are arguably correct — "the Committee saw no need for tightening" labeled Neutral but Claude says Dovish is a defensible reading. This highlights a broader issue: the dataset's annotations reflect one team's interpretive framework, and moderate inter-annotator disagreement is expected on hedged central bank text. Claude's Cohen's Kappa of 0.479 (moderate agreement) may partly reflect genuine annotation ambiguity rather than pure model failure.

5.4 Challenges & Mitigations

Overfitting. FinBERT overfits after epoch 3 on the small training set (1,683 sentences), visible in the diverging training/validation loss curves (Figure 3). Early stopping with the best validation checkpoint would have mitigated this, but Colab's 15GB storage could not retain 7–8GB checkpoints. The final-epoch model was used instead, likely understating FinBERT's best performance by a small margin.

Class imbalance. The ~50% Neutral majority could bias models toward conservative predictions. We mitigated this through macro-averaged F1 (weighting all classes equally) and stratified splits (maintaining class proportions across train/val/test). Class-weighted loss during training could further improve minority class recall.

Storage constraints. Each checkpoint required 7–8GB, exceeding Colab's free tier. We accepted the trade-off of using final-epoch models. This is a practical limitation of free-tier cloud computing that is worth noting for reproducibility.

5.5 Future Work

Several directions could push beyond the results observed here:

- 5.5.1 pre-training a model specifically on central bank text (FOMC transcripts, ECB speeches, BIS reports) to create a domain model with the right register
- 5.5.2 document-level context rather than isolated sentence classification, which would help resolve the multi-step inference and pragmatic implicature failures identified in Section 5.2
- 5.5.3 class-weighted loss to improve Dovish/Hawkish recall
- 5.5.4 exhaustive hyperparameter search for FinBERT
- 5.5.5 ensembling FinBERT and RoBERTa, which make different errors on 111 of 496 test sentences and could therefore complement each other.

6 Conclusion

We compared FinBERT (110M, financial domain) with RoBERTa-large (355M, general-purpose) on FOMC stance classification, replicating Shah et al. (2023). RoBERTa wins by 7–8 points (F1: 0.710 vs 0.633); both beat Claude’s zero-shot baseline (0.658). The mechanism is a domain mismatch: FinBERT’s corporate-finance pre-training does not transfer to central bank communication, which uses a fundamentally different register. For financial NLP, the source register of pre-training matters as much as whether it is “financial.”

7 References

Shah, A., Paturi, S., & Chava, S. (2023). Trillion Dollar Words: A New Financial Dataset, Task & Market Analysis. ACL 2023, pp. 6664–6679.

Yang, Y., Uy, M. C. S., & Huang, A. (2020). FinBERT: A Pretrained Language Model for Financial Communications. arXiv:2006.08097.

Liu, Y. et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Devlin, J. et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers. NAACL-HLT 2019.

Araci, D. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv:1908.10063.