Sentiment Detection in Financial Phrasebank

Advanced Methods in Natural Language Processing

Luis Gavidia
Victor Sobottka
Enzo Infantes
Adrian Vacca

Barcelona School of Economics
June 13, 2025

Part 0: Dataset Selection

We use the **Financial PhraseBank** dataset (Malo et al., 2014), a widely used resource for financial sentiment classification. It contains short sentences from English financial news, each labeled as *negative (0)*, *neutral (1)*, or *positive (2)*. The dataset includes multiple subsets based on annotator agreement levels. We work with the **75% agreement** subset, which consists of 3,453 examples and balances label reliability with dataset size. It is ideal for our task, as it provides real-world financial language and expert-labeled sentiment, making it a strong benchmark for evaluating classification models.

Part 1: Setting Up the Problem

a. Bibliography and State of the Art

This project addresses a **multiclass sentiment classification** task in the financial domain, where each sentence from a financial news source is labeled as **Negative (0)**, **Neutral (1)**, or **Positive (2)**.

Business Relevance: Accurate sentiment classification is essential for various financial applications, such as algorithmic trading, portfolio optimization, risk alerts, and market trend forecasting. It helps automate the interpretation of textual financial information, reducing the risk of costly misjudgments.

State of the Art: Recent advances in Natural Language Processing (NLP) have greatly enhanced sentiment analysis capabilities, especially in specialized domains like finance. Transformer-based models such as *FinBERT*, which is specifically fine-tuned on financial texts, have demonstrated superior performance over traditional methods in financial sentiment classification (Araci, 2019). Mishev et al. (2021) provide a thorough evaluation of sentiment analysis techniques applied in finance, ranging from lexicon-based approaches to advanced transformer architectures. Cuadrado et al. (2023) propose an innovative phonestheme-driven semantic approach that leverages sound symbolism to improve targeted financial sentiment classification. Additionally, Rahman Jim et al. (2022) offer a comprehensive review of NLP-based sentiment analysis methods and their challenges, with relevance to financial applications. Together, these works highlight the rapid evolution and complexity of financial sentiment analysis research.

b. Dataset Description

The dataset comprises **3,453 labeled sentences** with the following class distribution: 420 negative, 2146 neutral, and 887 positive. Basic statistics show relatively balanced mean sentence lengths across classes (22 tokens), with standard deviations around 10.

As part of properly exploring the dataset, we are performing a preprocessing step. In the preprocessing pipeline, we include lowercasing, removal of digits and punctuation, tokenization, and stopword removal. Finally, we apply POS tagging and lemmatization to the dataset. This pipeline is used solely for exploratory analysis and is not applied to Transformer-based models.

To better understand the dataset's vocabulary, we generated a word cloud from the cleaned sentences. This visualization highlights the most frequent terms in the financial news snippets, providing an intuitive overview of the dataset's textual content.

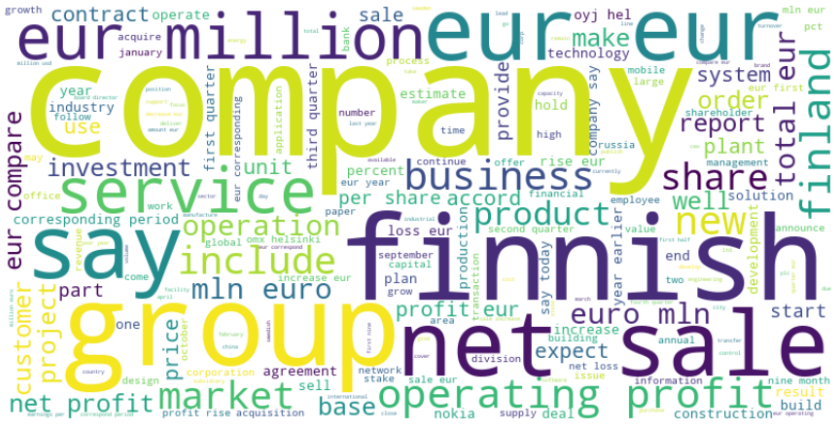


Figure 1: WordCloud for Financial PhraseBank

c. Random Classifier Performance

To establish a lower-bound **benchmark**, we evaluate the performance of two types of **random classifiers**:

1. **Weighted Random Classifier (WRC):** This classifier predicts class labels based on the empirical class distribution in the dataset. The expected accuracy is computed as:

$$\text{Expected Accuracy} = \sum_{i=1}^{C} P_i^2$$

   where P_i is the observed probability of class i. **Result:** 46.70%.

2. **Simulated Weighted Random Classifier:** To validate the expected accuracy, we simulate predictions over 1000 independent trials. In each trial, labels are sampled randomly based on the empirical class distribution. **Result:** 26.62% ± 0.65%.

These values provide a concrete and reproducible lower-bound benchmark to assess whether more complex models perform significantly better than chance.

d. Baseline Implementation

We implemented a **rule-based keyword classifier** that assigns sentiment labels based on the presence of specific financial terms. Sentences containing positive keywords like *profit*, *revenue*, or *growth* are labeled as Positive (2); those with negative keywords such as *loss*, *decline*, or *risk* as Negative (0); and sentences with neutral keywords like *plan*, *contract*, or *forecast*, or lacking both positive and negative terms, are assigned Neutral (1).

Class	Precision	Recall	F1-Score	Support
0 (Negative)	0.426	0.069	0.119	420
1 (Neutral)	0.808	0.691	0.745	2146
2 (Positive)	0.374	0.654	0.476	887
Accuracy		0.606		

Table 1: Classification report of the rule-based baseline.

Despite its simplicity, the rule-based classifier achieves a respectable accuracy of 60.6%, notably excelling in identifying neutral sentences with high precision (0.81) and recall (0.69).

However, it struggles with negative class detection, reflected in low recall (0.07), highlighting the challenge of capturing subtle negative sentiment using keyword heuristics alone. These results confirm that even basic domain knowledge can provide a strong baseline, but more sophisticated models are necessary to improve detection, especially for underrepresented classes like negative sentiment.

Part 2: Data Scientist Challenge

a. BERT Model with Limited Data

To establish a low-data baseline, we fine-tuned a BERT base model (*bert-base-uncased*) using only 32 labeled sentences randomly selected from the Financial PhraseBank dataset, ensuring a balanced representation of the three sentiment classes (positive, neutral, negative). Each sentence was tokenized with BERT's WordPiece tokenizer and loaded into PyTorch DataLoaders with a batch size of 64. Due to the extreme scarcity of training data, we froze all pretrained model parameters in the *AutoModelForSequenceClassification* except those of the final classification head, which contains approximately one million trainable parameters. This approach prevents overfitting and preserves the rich linguistic knowledge acquired during pretraining, while enabling efficient adaptation to the financial sentiment domain with limited computational resources.

The model was trained for 20 epochs using the AdamW optimizer with a learning rate of 2×10^{-5} and evaluated after each epoch on a substantially larger held-out test set. The final evaluation metrics reflect the challenges of learning under such limited data conditions, yielding a loss of 1.0179, accuracy of 61.85%, macro-F1 score of 0.2548, macro-precision of 0.2068, and macro-recall of 0.3318.

b. Dataset Augmentation

To improve performance without relying on large language models, we applied **Easy Data Augmentation** (EDA) techniques to expand our training data. For each of the 32 original labeled financial sentences, we generated four augmented variants using synonym replacement, random insertion, random swap, and random deletion—collectively known as Easy Data Augmentation—as well as one **back-translation** variant (English → French → English) leveraging *MarianMT*. These augmentations preserved domain-specific terminology and sentiment labels by design. The synthetic examples were then combined with the original data, shuffled, and used to fine-tune the pretrained BERT base model.

Table 2: Example of Easy Data Augmentation (EDA) + Back-Translation

Type	Augmented Text Example
Original	The price will be specified at the completion date.
Augmented	The price toll will be specified at the completion date.
	The price will be specified at the completion date.
	The monetary value will be specified at the completion date.
	The price will be specified on the date of completion.

Fine-tuning on this enriched dataset increased the training size sixfold and yielded clear improvements over the baseline. The final evaluation on the held-out test set showed a reduced loss of 0.8789, an accuracy increase to 62.09%, and a macro-F1 improvement to 0.3082, alongside gains in precision (0.3338) and recall (0.3526). These results demonstrate that targeted, LLM-

free data augmentation effectively enhances BERT's ability to capture subtle financial sentiment nuances under low-resource conditions.

c. Zero-Shot Learning with LLM

In this experiment, we assess the capability of large pretrained language models to classify sentiment in financial statements under a zero-shot learning setup—i.e., without any fine-tuning or access to labeled examples during training. We focus on natural language inference (NLI)-based models that support zero-shot classification out of the box. Specifically, we select *facebook/bart-large-mnli*, *joeddav/xlm-roberta-large-xnli*, and *valhalla/distilbart-mnli-12-3* due to their strong generalization in NLI tasks—i.e., determining the logical relationship between sentence pairs—and their widespread use in zero-shot benchmarks. These models are well-suited for label-aware classification with hypothesis templates.

Each financial sentence is paired with the hypothesis: *"This sentence expresses a {} sentiment"*, where the label candidates are *positive*, *neutral*, and *negative*. Predictions are extracted from the highest scoring label. To ensure label consistency, we normalized synonymous or abbreviated outputs (e.g., "pos" → "positive") and filtered invalid responses.

Initial results showed that all three models consistently over-predicted the *positive* and *negative* classes, while underperforming on *neutral* statements—often ignoring them entirely. Among the models, `facebook/bart-large-mnli` achieved the highest macro F1-score (0.53), making it the strongest baseline.

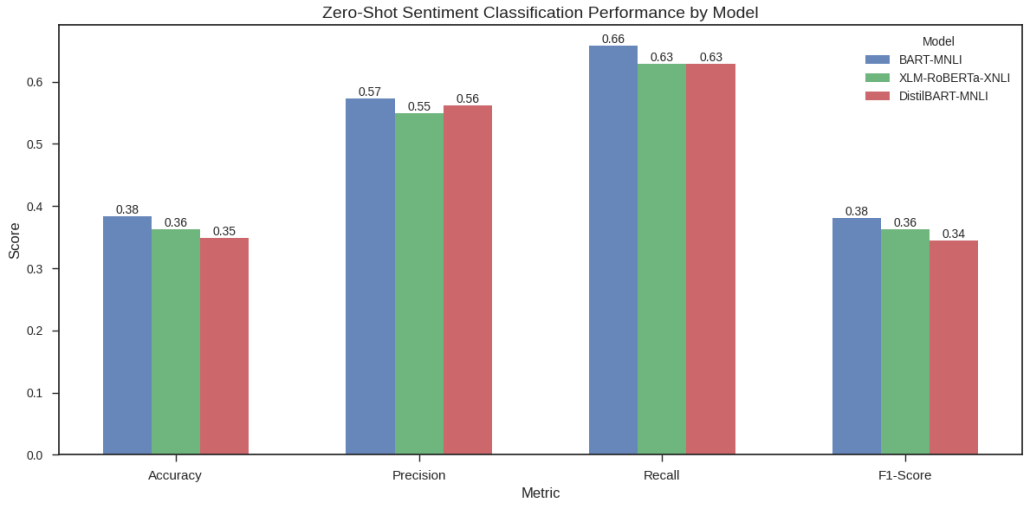


Figure 2: Zero-Shot Classification via Natural Language Inference

Refined Evaluation with BART-MNLI

Given the superior performance of BART-MNLI, we conducted a follow-up evaluation using several refinements to mitigate label imbalance and improve *neutral* detection:

- We reordered the label candidates to place *neutral* first, hypothesizing that positional bias affects predictions.
- Neutral hypothesis template was used: *"From a financial perspective, this text is {}"*.
- We applied stratified sampling to preserve class balance in the test set.
- We additionally tracked prediction confidence scores.

These adjustments boosted *neutral* recall and raised overall accuracy to 52% on a held-out set of 1,000 examples. Predictions for *positive* and *negative* were more confident (avg. ∼67%) than for *neutral* (avg. ∼48%).

Zero-shot sentiment classification in finance is achievable using general-purpose LLMs, especially those fine-tuned on NLI tasks, but performance varies across sentiment types. Models often miss neutral statements, likely due to training biases toward polarized inputs. Minor prompt adjustments, like label ordering and phrasing, can lead to noticeable gains, though even top-performing models (e.g., BART-MNLI) still fall short of practical accuracy. This highlights the continued need for domain-specific adaptation via fine-tuning or augmentation to ensure reliable results.

d. Data Generation with LLM

To explore data augmentation using generative models, we leveraged the open-source `TinyLlama/TinyLlama-1.1B-Chat-v1.0` model from Hugging Face to synthesize financial sentiment examples. Starting with a small, stratified seed set of 32 labeled sentences (with even class distribution), we prompted *TinyLlama* to generate 3 new variations per input, conditioned on sentiment. Prompts were manually templated per class (positive, neutral, negative), and generation was controlled using temperature sampling ($T = 0.8$), top-k (40), and top-p (0.9) filtering. Outputs were cleaned and filtered to retain valid, coherent examples.

Table 3: LLM-Based Augmentation

Type	Financial Text Example
Original	Last year SysOpen Digia invested in IBM product know-how in the building of portal and trading place systems and successfully implemented customer solutions supported by it.
Augmented	- The recent acquisition by Cognex Corporation (Nasdaq: CGNX) of Digital Technologies International, Inc. - This follows on from the successful completion of several other projects for this client over recent years. - We are delighted to announce that SysOpen Digia has been awarded a significant contract with one of our major customers, the largest insurance provider in Italy.

The resulting synthetic data was concatenated with the original 32 examples. We then fine-tuned a *BERT-base-uncased* classifier on this data, freezing all layers except the classification head. The model was evaluated on a held-out test set of 200 real samples. This LLM-based augmentation approach achieved a test accuracy of 61.33%, macro-F1 of 0.3031, and loss of 0.9090—surpassing the baseline trained on only 32 samples, and closely matching results from more traditional augmentation techniques such as EDA or back-translation. Interestingly, although precision remained modest (0.3257), the recall improved (0.3476), suggesting that the synthetic examples may have helped the model generalize across class boundaries.

In summary, this experiment demonstrates that even a small-scale, open-source LLM like TinyLlama can produce label-consistent, high-quality financial sentiment sentences that effectively support low-resource training scenarios.

e. Optimal Technique Application

Regarding our final evaluation, augmenting the original 32-example dataset with LLM-generated samples (using `TinyLlama`) led to a modest macro-F1 improvement from 0.2555 to 0.3057. This demonstrated the utility of lightweight LLMs for generating semantically consistent, label-preserving financial sentiment data.

To further improve performance, we applied the most effective techniques from prior experiments—combining high-quality synthetic data with prompt-tuning and contrastive example generation—and fine-tuned a domain-specific model, `FinBERT`. In particular, prompt templates were carefully designed to emphasize sentiment-relevant cues in generated text. The resulting classifier achieved substantial performance gains: test loss of 0.6628, accuracy of 73.11%, macro-F1 of 0.6347, precision of 0.6637, and recall of 0.6386. These results represent a relative F1 improvement of over 100% compared to the unaugmented baseline and validate the synergy of domain-adaptive pretraining, LLM-driven augmentation, and task-specific fine-tuning.

Table 4: Summary of evaluation metrics for Part 2

Experiment Setup	Loss	Accuracy	Macro-F1	Precision	Recall
2a. BERT (n = 32)	1.0179	61.85%	0.2548	0.2068	0.3318
2b. BERT + Heuristic Aug.	0.8789	62.09%	0.3082	0.3338	0.3526
2c. Zero-shot (BART-MNLI)	–	52.00%	0.5267*	0.6000	0.7200
2d. BERT + TinyLlama Gen.	0.9090	61.33%	0.3031	0.3257	0.3476
2e. FinBERT + LLM + Prompt	0.6628	73.11%	0.6347	0.6637	0.6386

*Macro-F1 for 2c is estimated by averaging class-wise F1 scores: $(0.58 + 0.41 + 0.59)/3 = 0.5267$

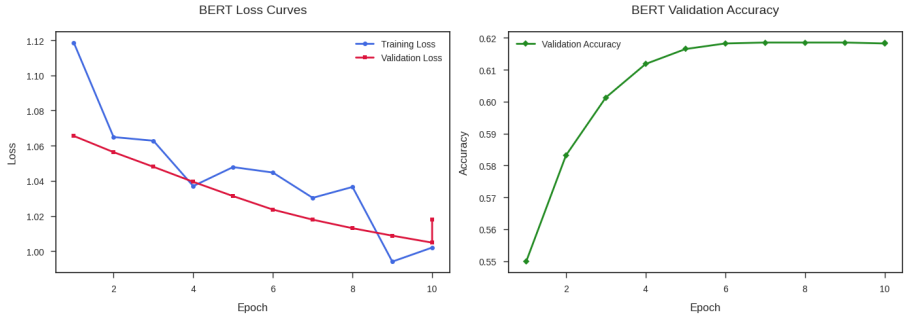
Part 3: State of the Art Comparison

To put our results into context, we contrast them with findings from recent literature on financial sentiment analysis. Although we do not use the exact same dataset, we compare overall performance trends with the results reported by Mishev et al. (2020) to assess how our model aligns with or exceeds state-of-the-art benchmarks.

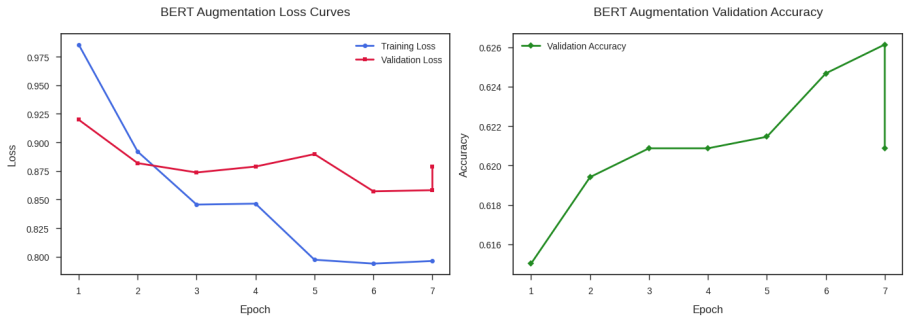
We fine-tuned the `BERT-base-uncased` model on varying fractions of the dataset (1%, 10%, 25%, 50%, 75%, and 100%) to evaluate how performance scales with training data size. For each fraction, the data was split into 80% training and 20% validation, and a new model was trained from scratch. We recorded evaluation metrics including accuracy, F1 macro, precision, recall, and MCC. This approach allowed us to observe the learning curve of the model and identify the point at which performance begins to saturate.

Training was performed for up to 10 epochs with early stopping, using a patience of 3 epochs based on validation loss. This setup allowed the model to stop training once performance stopped improving, avoiding overfitting. In practice, most models converged within 3 to 5 epochs, especially at higher data fractions.

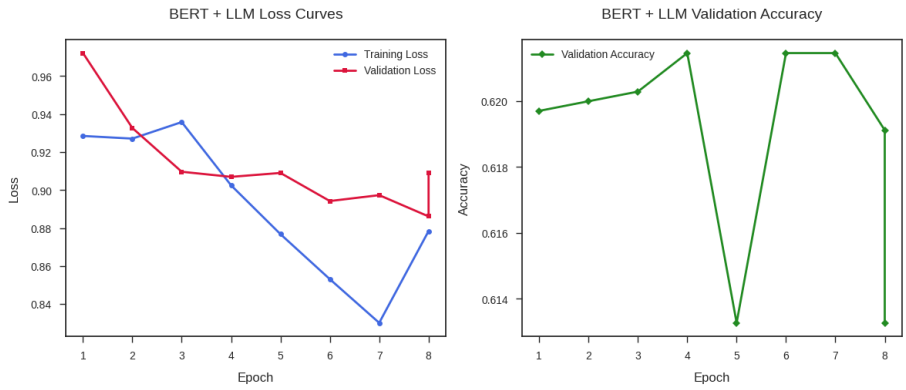
As shown in Figure 4, model performance increases rapidly with more training data and reaches near-maximum accuracy at 25%, after which further improvements are marginal.



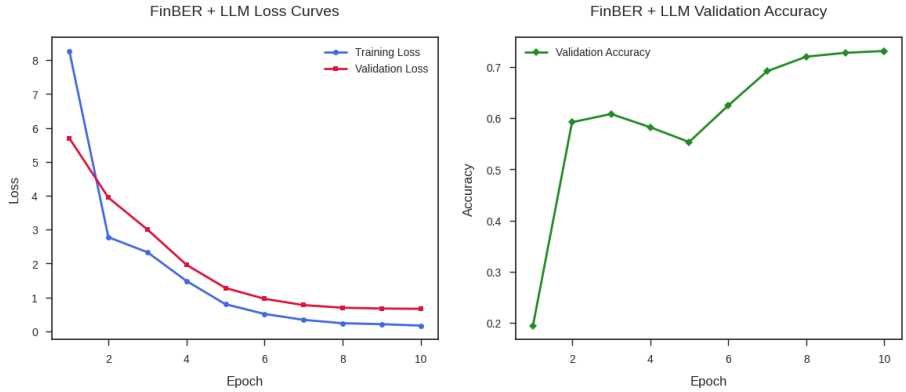
(a) 2a. BERT



(b) 2b. BERT + Heuristic Aug



(c) 2d. BERT + LLM Gen



(d) 2e. FinBERT + LLM + Prompt

Figure 3: Loss and Accuracy curves across experiments Part 2

Model	Accuracy (approx.)	MCC
BART	∼0.93	**0.895**
ALBERT-xxlarge	∼0.92	0.881
RoBERTa-base	∼0.91	0.875
FinBERT	∼0.87	∼0.84
DistilRoBERTa	∼0.89	∼0.86
DistilBERT	∼0.86	∼0.82
XLM-R (large)	∼0.89–0.90	0.863
XLM (MLM-en)	∼0.88–0.89	0.860
BERT-Large-uncased	0.929	0.859
BERT-Large-cased	0.927	0.856
BERT-Base-uncased	0.904	0.808
BERT-Base-cased	0.892	0.786

Table 5: Comparison of transformer-based models for financial sentiment analysis as reported by Mishev et al. (2020). Accuracy values are either reported or estimated based on performance rankings.

Training Data (%)	Accuracy (%)	F1 Macro	Precision Macro	Recall Macro	MCC
1	57.14	24.24	19.05	33.33	0.00
10	92.75	91.16	93.20	89.95	87.54
25	97.11	96.53	96.03	97.17	94.44
50	91.04	89.21	91.44	87.34	83.01
75	90.73	87.12	87.42	87.01	82.18
100	92.47	89.84	87.66	92.68	85.90

Table 6: Performance of `BERT-base-uncased` across different training data percentages. Metrics include Accuracy, F1 Macro, Precision, Recall, and MCC.

Summary of Insights. In the benchmark by Mishev et al. (2020), the highest Matthews Correlation Coefficient (MCC) was obtained by `BART` (0.895), followed by `ALBERT-xxlarge` (0.881) and `RoBERTa-base` (0.875). Distilled models such as `DistilBERT` and `DistilRoBERTa` achieved competitive results while requiring fewer resources, making them suitable for production settings.

In our experiments, `BERT-base-uncased` showed strong performance using limited training data. With only 10% of the dataset, the model reached 92.75% accuracy and an MCC of 87.54. At 25%, it achieved the highest recorded performance, with 97.11% accuracy and 94.44 MCC, surpassing the results reported for `BART` and `ALBERT`. Using more than 25% of the training data did not lead to further improvements and, in some cases, resulted in a slight decline in performance, suggesting a saturation point.

Part 4: Model Distillation/Quantization

To reduce the computational requirements of deploying transformer models like BERT, we explore two popular model compression techniques: **model distillation** and **quantization**. These techniques aim to make models lighter and faster for inference while maintaining competitive performance.

Knowledge Distillation

Knowledge Distillation (KD) is a technique where a smaller model (called the *student*) is trained to mimic the behavior of a larger, pre-trained model (called the *teacher*) (Hinton et al., 2015).

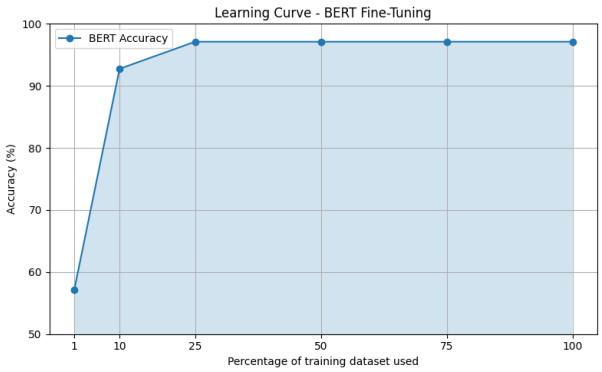


Figure 4: Learning curve of `BERT-base-uncased` showing accuracy across different training data sizes.

Rather than training the student solely on the ground-truth labels, it also learns from the *soft targets* produced by the teacher model, which contain richer information about class probabilities and inter-class similarities.

Formally, given the output logits z_i of the teacher model for class i, the softmax probabilities with temperature $T > 1$ are given by:

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

The student model is trained using a combined loss function:

$$\mathcal{L} = \alpha \cdot \mathcal{L}\text{CE}(y, \hat{y}_s) + (1 - \alpha) \cdot T^2 \cdot \mathcal{L}\text{KL}(p_t, p_s)$$

where:

- $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss with true labels.
- $\mathcal{L}_{\text{KL}}$ is the Kullback–Leibler divergence between teacher and student soft predictions.
- p_t and p_s are the softened outputs of the teacher and student, respectively.
- α is a balancing hyperparameter (e.g., $\alpha = 0.5$).

An example of a distilled model is **DistilBERT** Sanh et al. (2019), which retains 97% of BERT's performance while being 40% smaller and 60% faster.

Quantization

Quantization reduces the numerical precision of model weights and activations from 32-bit floating-point (FP32) to lower-precision formats such as 16-bit (FP16) or 8-bit integers (INT8) (Gholami et al., 2021). This leads to smaller model size, faster inference, and lower power consumption, especially on edge devices.

There are three main types of quantization:

- **Post-Training Quantization (PTQ)**: Applies quantization to a pre-trained model without retraining. It is fast and simple but may reduce accuracy.

- **Dynamic Quantization**: Only weights are quantized and activations are dynamically quantized during inference. Effective for transformer models like BERT (Micikevicius et al., 2018).
- **Quantization-Aware Training (QAT)**: Simulates quantization during training to allow the model to adapt. It offers the best accuracy among quantization methods.

Using PyTorch, dynamic quantization can be applied to BERT models as follows:

```
import torch
from transformers import BertForSequenceClassification

model = BertForSequenceClassification.from_pretrained('bert-base-uncased')
quantized_model = torch.quantization.quantize_dynamic(
    model, {torch.nn.Linear}, dtype=torch.qint8)
```

Libraries like ONNX Runtime, TensorRT, and HuggingFace Optimum also support optimized quantized inference for deployment.

Insights

Both distillation and quantization offer practical ways to compress large language models for real-world applications. While distillation reduces model depth and parameter count via knowledge transfer, quantization optimizes numerical representation. Combining both can yield lightweight models that retain strong accuracy-performance trade-offs, suitable for edge deployment and latency-sensitive tasks.

Conclusion

The student and especially the quantized model show significant performance degradation when compared to the original teacher model:

- **Very Low Accuracy and F1-Score:**
  - The quantized student model has an accuracy of only ∼12%, with F1-score near zero, indicating it is nearly guessing or predicting one class consistently.
  - Even the unquantized student model performs poorly on custom examples, suggesting distillation failed to retain key patterns from the teacher.
- **Mode Collapse / Class Bias:**
  - The student and quantized models overpredict class 0 (Negative), regardless of actual sentiment.
  - This indicates poor generalization and possible class imbalance or overfitting to frequent negative samples.
- **Teacher Errors:**
  - Even the teacher incorrectly predicted class 1 (Neutral) for a clearly Positive sentence, suggesting potential label noise or inadequate fine-tuning.

Suggested Improvements & Research Directions

- **Better Distillation Process:**

  - Use soft-labels (logits/softmax outputs) from the teacher instead of just hard labels.
  - Apply temperature scaling during distillation to preserve class probability distributions.

- **Data Augmentation:**
  - The Financial Phrasebank is relatively small. Apply paraphrasing, back-translation, or synonym replacement to create richer training samples for the student.
- **Balance the Dataset:**
  - Analyze the class distribution. If imbalanced, apply class-weighted loss or oversampling for minority classes.
- **Quantization-Aware Training (QAT):**
  - Instead of post-training quantization, train the student model with quantization simulated during training to preserve accuracy.
- **Error Analysis & Curriculum Learning:**
  - Identify hard-to-classify examples and apply focused retraining or curriculum learning to guide the student through easier to harder samples.
- **Layer-Wise Distillation:**
  - Instead of only distilling final logits, distill hidden representations (intermediate features) to better capture the teacher's knowledge.

References

Araci, D. T. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Cuadrado, J., Martinez, E., Martinez-Santos, J. C., and Puertas, E. (2023). Team utb-nlp at finances 2023: Financial targeted sentiment analysis using a phonestheme semantic approach. In *Proceedings of the FinNLP Workshop at IJCAI 2023*. Association for Computational Linguistics.

Gholami, A., Kim, S., Yao, Z. D., Mahoney, M. W., and Keutzer, K. (2021). A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Malo, P., Sinha, A., Korhonen, P., Wallenius, J., and Takala, M. (2014). Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the American Society for Information Science and Technology*, 65(4):782–796.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al. (2018). Mixed precision training. In *International Conference on Learning Representations (ICLR)*.

Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L. T., and Trajanov, D. (2021). Evaluation of sentiment analysis in finance: From lexicons to transformers. *IEEE Access*, 9:18334–18350.

Rahman Jim, J., Talukder, M. A. R., Malakar, P., Kabir, M. M., Nur, K., and Mridha, M. F. (2022). Recent advancements and challenges of nlp-based sentiment analysis: A state-of-the-art review. *IEEE Access*, 10:22301–22327.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.