Enhancing Financial Sentiment Analysis with Llama 3.1-8B - A Multi-Strategy Approach

Jonathan Domínguez * BBVA / Mexico

María de Jesús García † BBVA / Mexico

David Edgardo Castillo ‡ BBVA / Mexico

Abstract

This study addresses the challenges of financial sentiment analysis by leveraging advanced natural language processing techniques within large language models (LLMs). We fine-tune the Llama 3.1 model, incorporating instruction-based adjustments, a custom binary classification head, retrieval-augmented generation (RAG), and optimized prompting strategies. To ensure a robust evaluation, we unified multiple financial datasets, yielding a comprehensive training set that represents various financial contexts. Our results indicate that prompt optimization achieves an F1 score of 95% on the test set, while the custom binary classification head and RAG approaches each reach an F1 score of 92%. Instruction-based fine-tuning provides a moderate improvement with an F1 score of 80%, highlighting the additional value of model-specific enhancements. These findings underscore the potential for specialized NLP techniques to enhance financial sentiment analysis, supporting more informed and timely decision-making in financial markets. Code is available at here.

1 Introduction

In the rapidly evolving financial sector, sentiment analysis has become a critical tool for understanding market trends and making informed investment decisions. Financial sentiment analysis involves classifying textual data, such as news headlines, social media posts, and financial reports, into positive, negative, or neutral categories. Several classic machine learning models have been designed to solve financial sentiment classification (Li et al., 2014)(Fazlija and Harder, 2022). Though, this task presents unique challenges due to the complexity of financial texts, which often contain industry-specific jargon and numerical data. As a result, traditional machine learning approaches have struggled to keep pace with the growing need for accurate and real-time sentiment analysis. Recent advancements in natural language processing, particularly the advent of large language models (LLMs) like FinLlama (Konstantinidis et al., 2024), FinBERT (Araci, 2019) or Instruct-FinGPT (Zhang et al., 2023a), have shown significant promise in addressing these challenges. These models, when fine-tuned on domain-specific datasets, have demonstrated superior performance in various sentiment analysis tasks (Zhang et al., 2023c). The central hypothesis of this paper is that by implementing instruction fine tuning to the Llama 3.1-8B model (similar to the Instruct-FinGPT solution (Zhang et al., 2023a)), optimizing prompts, and incorporating retrieval-augmented generation (RAG) (Lewis et al., 2020), we can achieve incremental improvements in financial sentiment analysis performance. Furthermore, the novel idea introduced by FinLlama (Konstantinidis et al., 2024), which consists in adding a custom softmax layer at the output of the foundational model, can give a substantial boost in model performance; after fine tuning the model using this new binary classification training objective. In this case, we propose the addition of a linear layer and a softmax layer, which can result in an even greater predictive power. This hypothesis is built on the assumption that these techniques will enhance domain adaptation, contextual understanding, and sentiment categorization. We outline the steps taken to fine-tune a large language model, incorporate various optimization techniques using Low-Rank Adaptation (LoRA) (Hu et al., 2021), implement prompt optimization strategies, incorporate retrieval augmented generation (RAG), and compare the performance of our solutions against the baseline model results. The performance results are based on the binary classification objective, which consists on getting either a "positive" or "negative" sentiment.

* jonathan.dominguez@bbva.com
† mariadejesus.garcia.santiago@bbva.com
‡ davidedgardo.castillo@bbva.com

2 Related work

The use of sentiment analysis in predicting market trends has become increasingly prevalent as a tool for making informed investment decisions. Numerous studies leverage sentiment analysis to forecast stock trends and assess the impact of news on stock prices. However, predicting stock trends remains a complex task due to the need to address biases in sentiment models and consider external factors like inflation, as noted by (Fazlija and Harder, 2022). In their work, they implemented Sentiment-BERT with domain adaptation for financial data, which resulted in model performance below 57% accuracy. Similarly, (Bhat and Jain, 2024) reported low accuracy when using sentiment analysis alone for stock price predictions, finding that classical machine learning models (e.g., neural networks, random forests, and logistic regression) trained solely on sentiment-encoded news headlines using the emotion-english-distilroberta-base model performed worse than models trained only on financial data features. In contrast,(Li et al., 2014) found that incorporating a Bag-of-Words (BoW) approach, enhanced by reducing dimensionality with a sentiment-based vector space, improved performance over using BoW alone.

Beyond stock prediction, sentiment analysis has also been applied to corporate credit risk assessment. By analyzing financial news, researchers can gauge a company's credit risk at a specific time. (Ahbali et al., 2022) utilized entity sentiment analysis and multi-label classification, employing a pre-trained Electra transformer for this purpose. Similarly, (Tang et al., 2023) used entity sentiment analysis to create a financial sentiment dataset for fine-tuning NLP models, finding that a pre-trained BERT-CRF model fine-tuned on their dataset (FinEntity) outperformed ChatGPT in zero-shot and few-shot tasks for cryptocurrency sentiment detection. As (Ahbali et al., 2022) highlighted, data quality significantly impacts model performance. Based on these findings, we opted to incorporate FinEntity into our dataset to enhance our model's development.

Historically, sentiment analysis in finance has been dominated by transformers and small language models (SMLs) like T5. However, with the rise of large language models (LLMs) outperforming SMLs in various NLP tasks, it is pertinent to evaluate their efficacy in complex tasks like sentiment analysis. (Zhang et al., 2023c) showed that LLMs could surpass SMLs in sentiment analysis, answering key questions on LLM performance compared to SMLs in zero-shot and few-shot scenarios. Their research demonstrated that LLMs' few-shot context learning often outperformed fine-tuned SMLs, even when limited data was available. (Zhang et al., 2023a) also found that instruction fine-tuning further improved LLM performance (e.g., LLaMA-7B) compared to traditional transformers fine-tuned for financial sentiment classification. Consequently, we selected LLama 3.1 8B for our analysis, anticipating better performance over SMLs or conventional transformers, as evidenced by (Zhang et al., 2023a). Further details about LLama 3.1 will be discussed in Section 4.

While zero-shot LLMs often outperform SLMs, methods exist to enhance zero-shot results. One approach is instruction-based fine-tuning, as proposed by (Zhang et al., 2023a). Alternatively, modifying LLM structure, by adding a softmax layer with instruction-based fine-tuning, has set new benchmarks in financial sentiment analysis (Konstantinidis et al., 2024). A final approach involves combining instruction-based fine-tuning with Retrieval-Augmented Generation (RAG), a technique that incorporates external financial knowledge and achieves superior results compared to standard fine-tuning (Zhang et al., 2023b).

3 Data

We unified four financial sentiment analysis datasets into a single one, to perform fine tuning, prompt optimization and retrieval augmented generation.

- Twitter financial dataset [1] : This is an English-language dataset containing an annotated corpus of finance-related tweets. This dataset is used to classify finance-related tweets for their sentiment. The dataset holds 11,932 documents annotated with 3 labels:

  1. "LABEL_0": "Bearish",
  2. "LABEL_1": "Bullish",
  3. "LABEL_2": "Neutral".

  The data was collected using the Twitter API. The current dataset supports the multi-class classification task.

[1] https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment

- FiQA dataset (Malo et al., 2014) [2]: This challenge focuses on advancing the state-of-the-art of aspect-based sentiment analysis and opinion-based Question Answering for the financial domain with 485 samples.

- FinEntity (Ahbali et al., 2022) [3]: Entity-level sentiment classification dataset, called FinEntity, that annotates sentiment (positive, neutral, and negative) of individual financial entities in financial news. The dataset holds 979 news headlines annotated with 3 labels.

- Financial PhraseBank (Malo et al., 2014) [4]: This dataset contains the sentiments for financial news headlines from the perspective of a retail investor. The sentiment can be negative, neutral or positive. The dataset holds 4837 news headlines.

Since financial decision-making is often reduced to buying or selling assets, sentiment analysis of financial news should focus exclusively on "positive" and "negative" sentiments. Excluding "neutral" sentiments helps ensure that sentiment analysis more directly corresponds to either a long or short market position. Additionally, we observed some ambiguity in neutral phrases. For example, sentences with a positive tone were sometimes classified as neutral, likely due to a lack of context clarifying why these sentences should be considered neutral.

After removing all "neutral" sentiments from the unified dataset, the data was randomly split into three sets: 70% for training, 10% for validation, and 20% for test. A random seed was applied to ensure replicability, resulting in 5,105 samples for training, 1,445 samples for test, and 744 samples for validation.

4 The Llama Model

Llama 3.1 is a dense transformer model designed to predict the next token in a text sequence (Dubey et al., 2024). The model takes a sequence of tokens as input, embedding each token and passing them through multiple blocks composed of self-attention and feed-forward layers. The resulting output token is then used as the next input to generate subsequent tokens. Compared to previous versions, Llama and Llama 2, the primary architectural changes in Llama 3.1 include:

- Grouped query attention (GQA) with eight key-value heads, improving inference time and memory use in decoding.

- Modified attention mask to prevent self-attention between different documents within the same sequence

- The vocabulary size was changed to $128K$ tokens. The tokens are distributed: $100K$ from the tiktoken tokenizer for English and $28K$ for multilingual texts.

- Increasing RoPE base frequency hyperparameter to $500,000$ supports long input sequences.

In Section 2, we discussed that data quality and fine-tuning are critical factors in enhancing transformer and language model performance. Fine-tuning alone can significantly boost performance, but when paired with high-quality data, it achieves even better results (Ahbali et al., 2022). The Llama model benefits similarly from this approach. The authors describe two main stages in developing Llama models:

- Language model pre-training. The model is trained on a large multilingual text corpus; the texts are converted into discrete tokens and used to predict the next word in the sequence. This phase helps the model learn the language structure and acquire knowledge from the text.

- Language model post-training. The model has acquired knowledge of language structure but does not behave as intended for an assistant. Several rounds of human feedback changed their behaviour with supervised fine-tuning on instruction tuning data.

For our analysis, we utilized Llama 3.1 8B. The "8B" designation indicates that the model was trained with eight billion tokens. While larger versions (70B and 405B tokens) exist, they would require multiple GPUs merely to load for zero-shot predictions. To illustrate the differences in model scale, the 8B version comprises 32 layers, whereas the 70B and 405B versions include 80 and 126 layers, respectively.

[2] https://www.kaggle.com/datasets/sbhatti/financial-sentiment-analysis

[3] https://huggingface.co/datasets/yixuantt/FinEntity.

[4] https://www.kaggle.com/datasets/sbhatti/financial-sentiment-analysis.

In the following section, we will present our proposed methods to enhance Llama 3.1's performance using zero-shot techniques, establishing a baseline model. Our proposals include instruction fine-tuning, instruction fine-tuning with an additional softmax layer, RAG, and prompt optimization.

5 Methods and Experiments

We present the methodologies and experimental procedures implemented to enhance financial sentiment analysis performance using large language models. Building on the capabilities of Llama 3.1, we evaluate various approaches, including basic prompting, instruction fine-tuning, a custom binary classification head, retrieval-augmented generation (RAG), and prompt optimization. Additionally, we explore an ensemble model to aggregate predictions across multiple methods, providing a comprehensive analysis of performance metrics, including precision, recall, and F1 score, to validate each approach's effectiveness.

5.1 Basic prompting of baseline model

We used Llama 3.1 to classify each financial sentence as positive or negative. For each sentence, the model was prompted five times, with each prompt returning a categorical answer—positive, negative, or unknown. Repeating the prompt five times per sentence helped improve the model's precision and reduce the likelihood of hallucination. The final prediction was determined by majority voting on the five categorical answers (positive or negative) generated by the model.

5.2 Instruction Fine Tuning

We used financial training data to update the weights of the base model, adapting Llama to the financial context. Fine-tuning was conducted with the LoRA adapter to avoid the high computational cost of full fine-tuning. Since the model is used for text generation, labels were embedded to enable loss calculation for each training epoch. As in the previous approach, we prompted the fine-tuned model five times to determine the final prediction based on the mode of the responses.

5.3 Fine Tuning with Custom Binary Classification Head

This strategy builds on the previous one by adding a final softmax layer. With this addition, each prompt's output is a two-dimensional probability between 0 and 1, representing the likelihood of the financial sentence being positive or negative. The final prediction is determined by the higher of the two scores. Again, during training the LoRA adapter was used to avoid high computational costs. Although we did not implement majority voting in this approach, we believe it could further improve the model's performance.

5.4 Retrieval Augmented Generation

In this strategy, we implemented a retrieval mechanism that provides three contextual passages to each prompt given to Llama 3.1. The retrieval context data consists of passages from Wikipedia, obtained through the training dataset. Specifically, we extracted all capitalized words from the training set, searched each word on Wikipedia, and stored the resulting passages. This approach is based on the assumption that capitalized words are likely to be the most relevant in each sentence. Once the context data was gathered, we encoded it using the pre-trained ColBERT 2.0 model from Python's ragatouille package [5].

For each financial sentence given to Llama 3.1, three context passages are retrieved via ColBERT, so the final prompt includes both the financial sentence and the retrieved context. As in the previous strategies, the final prediction is determined by the majority voting of the five prompts given to Llama 3.1.

5.5 Prompt optimization of Baseline Model

In this approach, we used DSPy (Khattab et al., 2022, 2023) to optimize the prompts passed to Llama 3.1. DSPy is a framework designed to optimize fine-tuning for language model prompting. One advantage of DSPy is that it allows you to start with a base prompt and refine it through algorithms, reducing the time needed to tailor prompts to specific data. DSPy includes modules that define the basic task for the language model and optimizers that search for the optimal prompt based on a selected metric. We chose DSPy to compare the results of optimized prompting with those from traditional prompting techniques.

Our prompt optimization strategy comprises two signature modules containing prompts that are optimized during training. The first module is a

[5] https://github.com/run-llama/llama_index/tree/main/llama-index-packs/llama-index-packs-ragatouille-retriever

`ContextSignature`, where we provide the language model with the instruction for the binary classification task described earlier. The second module is a `Parser`, which interprets the output from the `ContextSignature` in the desired format, aiming to reduce hallucinations in the language model's output.

In the next code block we describe the `ContextSignature` with the format used by DSPy. As we mentioned, the `__doc__` describe the task for the LM, the `dspy.InputField` pass the description of the documents that are the input for the LM and `dspy.OutputField` has the description of the desired output. We included a `BootstrapFewShot` optimizer to optimize the program using few shots and a custom metric.

```
class ContextSignature(dspy.Signature):
    __doc__ = """Classify the financial nmews healines
    in the given categories

    The categories are given as:

    'positive': A headline that suggest good news or
    positive developments.
    'negative': A headline that suggest downturns, losses,
    challenges or negative developments.
    """

    news = dspy.InputField(desc="Financial news headlines.")
    classification = dspy.OutputField(desc="ONLY write the word
    'positive' or 'negative', nothing else!")
```

5.6 Ensemble of Predictions

We gathered the final predictions from the Baseline Model, the Instruction Fine-Tuning Model, and the Fine-Tuning Model with a Custom Binary Classification Head. The ensemble method used is hard voting, meaning that for each news headline, the final prediction is determined by the majority class.

5.7 Metrics

We used the following metrics to train, validate and test all the aforementioned approaches:

- Precision: This metric is computed for each class as the number of the correctly predicted cases divided by the number of all predicted cases of the given class. For each class, this metric summarizes the percentage of correct predicted cases among all the predictions of the given class.

$$Precision = \frac{TP}{TP + FP}$$

.

- Recall: This metric is computed for each class as the number of the correctly predicted cases divided by the number of cases of that class. For each class, this metric summarizes the percentage of correct predicted cases among all the true values of the given class.

$$Recall = \frac{TP}{TP + FN}$$

.

- Macro F1 score: Macro F1 is a macro-averaged F1 score. To calculate macro F1, two different averaging-formulas have been used: the F-score of (arithmetic) class-wise precision and recall means or the arithmetic mean of class-wise F-scores, where the latter exhibits more desirable properties.

$$F_1 score = 2 * \frac{precision * recall}{precision + recall}$$

6 Results and Analysis

Our experiments revealed significant performance variations across the different approaches employed to enhance financial sentiment classification using the Llama 3.1 model. Table 1 summarizes the comparative results for each approach.

Prompt Optimization achieved the best overall results, reaching an F1 score of 95% on the test set, thereby establishing itself as the most effective method for precise sentiment interpretation. This method's success highlights the importance of well-tuned prompts in reducing model ambiguity, particularly in finance, where language can be highly nuanced.

The Custom Binary Classification Head (CBH) and Retrieval Augmented Generation (RAG) methods also performed well, each obtaining an F1 score of 92%. CBH's specialized classification head contributed to high precision in distinguishing positive and negative sentiments, while RAG's use of additional context via retrieved passages enriched the model's understanding in cases requiring nuanced interpretation. Despite RAG's advantages, its dependency on the quality of retrieved content reveals a potential area for improvement, as inaccuracies in external information could affect model outputs.

The Instruction Fine-Tuning (IF) approach offered a moderate enhancement over the baseline, with an F1 score of 80%. While this indicates that fine-tuning with domain-specific data can elevate performance, the results suggest that fine-tuning

alone may not capture the full complexity of financial sentiment, especially without targeted prompt adjustments or classification head modifications.

The ensemble model approach—combining predictions from the Baseline, IF, and CBH methods—achieved a strong F1 score of 92%, indicating the robustness of integrating predictions across models. However, it did not surpass prompt optimization, underscoring that specialized single-model adjustments can offer higher precision.

In summary, the results underscore the adaptability of Llama 3.1 in financial sentiment analysis and the effectiveness of model-specific enhancements. While each technique has unique strengths, prompt optimization and custom classification layers were particularly valuable for achieving high performance. These findings suggest that, with refined prompt techniques and targeted adjustments, Llama 3.1 is highly suited to the nuanced demands of financial sentiment classification.

	Method	Precision	Recall	F1
Train	BM	80%	81%	80%
	IF	83%	85%	82%
	CBH	**95%**	93%	**94%**
	RAG	92%	**94%**	93%
	OP	92%	93%	92%
	Ensemble	92%	93%	92%
Test	BM	78%	80%	78%
	IF	82%	83%	80%
	CBH	93%	91%	92%
	RAG	91%	92%	92%
	OP	**95%**	**95%**	**95%**
	Ensemble	91%	93%	92%
Val	BM	82%	84%	82%
	IF	82%	84%	81%
	CBH	95%	92%	93%
	RAG	91%	92%	92%
	OP	**96%**	**96%**	**96%**
	Ensemble	91%	92%	91%

Table 1: Results comparison of the five approaches. 1) Baseline Model (BM), 2) Instruction Fine Tuning (IF), 3) Fine Tuning with Custom Binary Classification Head (CBH), 4) Retrieval Augmented Generation (RAG) and 5) Prompt optimization of Baseline Model (PO)

Known Project Limitations

The first limitation is the binary classification approach, which excluded the neutral class due to ambiguous phrases identified during exploratory analysis. For example, sentences with a positive tone were sometimes classified as neutral, possibly because additional context was needed to clarify why these sentences were considered neutral.

The retrieval mechanism also has limitations, as it can only search for words that directly match Wikipedia URLs. This could be improved by refining the dictionary of words. For instance, the company "Apple" appears in financial sentences simply as "Apple," but to search for the company in Wikipedia (and not the fruit), we need to use "Apple_Inc." Additionally, the passages retrieved from Wikipedia may match terms in our dictionary but don't necessarily provide financial context specific to the companies mentioned.

7 Conclusion

This study demonstrates that while Llama 3.1 achieves reasonable performance in binary financial sentiment classification, its accuracy and adaptability improve significantly with advanced techniques such as Prompt Optimization, Custom Binary Classification Head (CBH), and Retrieval Augmented Generation (RAG). Among these, prompt optimization delivered the highest precision, recall, and F1 scores, underscoring the importance of refined prompt structures in enhancing model performance for specialized tasks.

Overall, this work contributes to the field by demonstrating that prompt optimization and targeted model enhancements can significantly elevate sentiment analysis performance in financial applications. For future research, refining prompt optimization strategies, enhancing retrieval quality for RAG, and exploring hybrid models that dynamically adapt to sentiment context may offer additional avenues for advancing financial sentiment analysis accuracy. This work not only highlights effective approaches for current models but also establishes a foundation for ongoing improvements in financial sentiment classification as natural language processing techniques continue to evolve.

Authorship Statement

In this work, Jonathan primarily contributed to the implementation of the Baseline Model (BM) and the Custom Binary Head classification (CBH). Maria focused on the Instruction Fine-Tuning (IF) and Prompt Optimization (PO) implementations, while David was mainly responsible for the Retriever Augmented Generation (RAG) implementa-

tion and data preparation. All other tasks, including the literature review, were completed in equal proportion.

References

Noujoud Ahbali, Xinyuan Liu, Albert Nanda, Jamie Stark, Ashit Talukder, and Rupinder Paul Khandpur. 2022. Identifying corporate credit risk sentiments from financial news. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 362–370.

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

Rithesh Bhat and Bhanu Jain. 2024. Stock price trend prediction using emotion analysis of financial headlines with distilled llm model. In *Proceedings of the 17th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 67–73.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Bledar Fazlija and Pedro Harder. 2022. Using financial news sentiment for stock price direction prediction. *Mathematics*, 10(13):2156.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Thanos Konstantinidis, Giorgos Iacovides, Mingxue Xu, Tony G Constantinides, and Danilo Mandic. 2024. Finllama: Financial sentiment classification for algorithmic trading applications. *arXiv preprint arXiv:2403.12285*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. 2014. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23.

Pekka Malo et al. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Yixuan Tang, Yi Yang, Allen H Huang, Andy Tam, and Justin Z Tang. 2023. Finentity: Entity-level sentiment classification for financial texts. *arXiv preprint arXiv:2310.12406*.

Boyu Zhang, Hongyang Yang, and Xiao-Yang Liu. 2023a. Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models. *arXiv preprint arXiv:2306.12659*.

Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023b. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 349–356.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023c. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

A Example Appendix

A.1 Basic prompting of baseline model

		Precision	Recall	F1
Train	Positive	69%	88%	77%
Train	Negative	91%	75%	82%
Train	Macro avg	80%	81%	80%
Test	Positive	67%	87%	76%
Test	Negative	90%	73%	80%
Test	Macro avg	78%	80%	78%
Val	Positive	71%	92%	80%
Val	Negative	94%	76%	84%
Val	Macro avg	82%	84%	82%

Table 2: Base line model

A.2 Instruction Fine Tuning

		Precision	Recall	F1
Train	Positive	70%	96%	81%
Train	Negative	97%	73%	83%
Train	Macro avg	83%	85%	82%
Test	Positive	67%	96%	79%
Test	Negative	96%	70%	81%
Test	Macro avg	82%	83%	80%
Val	Positive	69%	95%	80%
Val	Negative	96%	73%	83%
Val	Macro avg	82%	84%	81%

Table 3: Instruction Fine Tuning

A.3 Fine Tuning with Custom Binary Classification Head

		Precision	Recall	F1 Score
Train	Positive	99%	86%	92%
Train	Negative	92%	99%	95%
Train	Macro avg	95%	93%	94%
Test	Positive	97%	83%	89%
Test	Negative	90%	98%	94%
Test	Macro avg	93%	91%	92%
Val	Positive	98%	85%	91%
Val	Negative	91%	99%	95%
Val	Macro avg	95%	92%	93%

Table 4: Binary Classification Head

A.4 Retrieval Augmented Generation

		Precision	Recall	F1
Train	Positive	88%	96%	92%
Train	Negative	97%	91%	94%
Train	Macro avg	92%	94%	93%
Test	Positive	86%	95%	90%
Test	Negative	96%	90%	93%
Test	Macro avg	91%	92%	92%
Val	Positive	85%	96%	90%
Val	Negative	97%	89%	93%
Val	Macro avg	91%	92%	92%

Table 5: RAG

A.5 Prompt optimization of Baseline Model

		Precision	Recall	F1
Train	Positive	95%	96%	96%
Train	Negative	97%	97%	97%
Train	Macro avg	96%	97%	97%
Test	Positive	93%	95%	94%
Test	Negative	97%	95%	96%
Test	Macro avg	95%	95%	95%
Val	Positive	95%	96%	95%
Val	Negative	97%	96%	97%
Val	Macro avg	96%	96%	96%

Table 6: Prompt Optimization

A.6 Ensemble of Predictions

		Precision	Recall	F1
Train	Positive	87%	96%	91%
Train	Negative	97%	91%	94%
Train	Macro avg	92%	93%	92%
Test	Positive	86%	96%	91%
Test	Negative	97%	90%	93%
Test	Macro avg	91%	93%	92%
Val	Positive	84%	96%	90%
Val	Negative	97%	89%	93%
Val	Macro avg	91%	92%	91%

Table 7: Ensemble