Leveraging LLM Ensembles for Robust Sentiment Classification Computational Intelligence Lab Project Report

Afonso Ferreira da Silva Domingues [*1] **Rahul Kaundal** [*1] **Thösam Norlha-Tsang** [*1]

Group: *Siuuupremacy*

Abstract

Sentiment classification refers to the automated categorization of unstructured text into sentiment-based classes, typically binary, ternary, or fine-grained numerical categories. As a foundational task in natural language processing, it has received extensive attention in both research and industry due to its broad range of applications. This project addresses a ternary classification problem, where each text sample must be labeled as positive, neutral, or negative. In this report, we compare classical machine learning models with transformer-based architectures, examining the effects of preprocessing strategies, model ensembling, and robustness enhancements. Our goal is to highlight the trade-offs between model complexity, performance, and practical deployability.

Keywords: Sentiment Analysis, Natural Language Processing, Ensembles, Large Language Models

1. Introduction

Sentiment classification is a natural language processing task that involves assigning predefined sentiment labels to unstructured text. It plays a central role in a variety of real-world applications, including product review analysis, customer feedback monitoring, social media opinion mining, and market research. Despite its long-standing history, sentiment classification remains a challenging problem due to the inherent complexity of natural language, the noise and variability in real-world data, and the subtlety of subjective expressions, including sarcasm, ambiguity, and informal language.

Early sentiment classification methods relied heavily on classical machine learning models such as Logistic Regression, Support Vector Machines, and Random Forests, which typically required extensive preprocessing and handcrafted features to perform well. With the emergence of distributed word representations and deep learning architectures, particularly the introduction of pretrained language models like BERT, the field has seen a shift toward models that learn contextual representations directly from raw text. These models have demonstrated significant improvements in performance and robustness, but they also introduce challenges related to computational cost and interpretability.

In this project, we explore sentiment classification in a ternary setting (positive, neutral, negative) using a labeled dataset of over 100,000 sentences. We compare classical machine learning models with a variety of state-of-the-art transformer-based encoders. Beyond model performance, we investigate the effects of preprocessing, ensemble methods, and robustness strategies such as LLM-generated paraphrasing. Through this evaluation, we aim to provide insights into the trade-offs between model complexity, performance, and practical deployment.

2. Related Work

The first studies on sentiment classification employed classical machine learning models such as Naive Bayes, Support Vector Machines (SVM), and Maximum Entropy classifiers (Pang et al., 2002; Go et al., 2009). These models typically relied on bag-of-words representations and handcrafted features, which often required extensive preprocessing and domain-specific tuning. While computationally efficient, their effectiveness was limited by their inability to capture contextual information.

The introduction of contextual word embeddings like Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) enabled models to learn semantic relationships between words, paving the way for neural architectures (Kim, 2014; Socher et al., 2013). More recently, transformer-based models have become state-of-the-art for sentiment analysis by leveraging contextual word representations from large-scale pretraining. These models have significantly

[*]Equal contribution [1]Department of Computer Science, ETH Zürich, Switzerland. Correspondence to: Afonso Ferreira da Silva Domingues <aferreira@ethz.ch>, Rahul Kaundal <rkaundal@ethz.ch>, Thösam Norlha-Tsang <tnorlha@ethz.ch>.

Computational Intelligence Lab FS2025, ETH Zürich

improved accuracy and robustness but also introduced new challenges, primarily related to high resource demands (Devlin et al., 2019).

3. Data

3.1. Data Exploration

The dataset used for sentiment classification was partitioned into three subsets: training (80%), validation (10%), and test (10%). Each subset is created using stratified sampling to maintain the original class distribution. The training set comprises 17,528 negative (21.46%), 39,318 neutral (48.14%), and 24,831 positive (30.40%) examples. Both the validation and test sets follow the same distribution, with 2,191 negative, 4,915 neutral, and 3,104 positive samples each. This class imbalance, where the neutral class dominates, highlights the need for strategies that mitigate bias during model training and evaluation.

Additionally, during preliminary analysis, we observed that a small portion of the dataset contains sentences written in various languages such as French, Dutch, and German. These samples represent a minority and do not significantly affect the overall distribution. In addition, some entries consist of noise rather than meaningful text. For instance, isolated URLs, symbols, or malformed strings, which do not convey actual sentiment or review content.

3.2. Data Preprocessing

Different preprocessing strategies were applied depending on the type of model. For classical machine learning models such as Logistic Regression, Random Forest, XGBoost, and Multi-Layer Perceptron, we performed several text normalization steps to reduce noise and standardize input. Specifically, we expanded contractions (e.g., ”don’t” → ”do not”), and removed email addresses and URLs. While stemming and lemmatization were also explored, they did not yield any performance improvements and were subsequently omitted. In contrast, for encoder-based models such as BERT, no preprocessing was applied. These models are pretrained on raw, unfiltered text that naturally includes contractions, email addresses, URLs, and other informal elements, and are therefore robust to such inputs without the need for additional cleaning.

To mitigate the effects of class imbalance during model training, we also explored data augmentation for the negative and positive classes, as well as downsampling for the neutral and positive classes. However, both strategies resulted in a decline in performance when applied to transformer-based models.

TODO: SEE IF YOU WANT TO KEEP THIS, I THINK IT IS GOOD BECAUSE IT SHOWS WE TESTED IT

4. Setup

4.1. Models

For the sentiment classification task, we evaluated a range of models, both classical machine learning approaches and modern transformer-based encoder models. The classical models included Logistic Regression, Random Forest, XGBoost, and Multi-Layer Perceptron. Among these, Logistic Regression achieved the best performance when trained on the raw, unprocessed data. Based on this, preprocessing was applied only to Logistic Regression to investigate whether cleaning and normalizing the text could further improve its results. The other models were kept on raw data since they did not outperform Logistic Regression without preprocessing.

On the other hand, for the encoder-based category, we fine-tuned several pretrained models from the Hugging Face Transformers library: BERT (base, cased) (Devlin et al., 2019), BERT (multilingual) (Devlin et al., 2019), DistilBERT (base, cased) (Sanh et al., 2020), DistilBERT (multilingual) (Sanh et al., 2020), RoBERTa (base) (Liu et al., 2019), XLM-RoBERTa (base) (Conneau et al., 2020), RoBERTa (large) (Liu et al., 2019), DeBERTa v3 (base) (He et al., 2023), and DeBERTa v3 (large) (He et al., 2023). These models vary significantly in their architectural complexity and number of parameters, ranging from lightweight variants like DistilBERT to large-scale models such as RoBERTa-large and DeBERTa v3-large. This variety allows for a detailed comparison between traditional and modern techniques in terms of accuracy, robustness, and computational efficiency.

We chose the cased variants of the models because case information can provide valuable cues for sentiment analysis. In addition, all models were fine-tuned using the default classification head for sequence classification provided by the `AutoModelForSequenceClassification` class. Future work could explore how different classification head architectures affect the performance and generalization of transformer-based sentiment classifiers.

TODO SHOULD WE KEEP THE SENTENCE ABOUT HOW WE DID NOT TEST PREPROCESSING WITH THE OTHER MODELS? ALSO, KEEP SENTENCE FUTURE WORK DIFFERENT CLASSIFICATION HEADS?

4.2. L_score Function

For evaluating the performance of the models we used the following function:

$$L(\hat{y}, y) = 0.5 \cdot \left(2 - \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|\right)$$

The scoring function penalizes severe misclassifications (e.g., positive confused for negative) more than minor misclassifications (e.g., positive confused for neutral), encouraging models that can accurately distinguish between contrasting emotional polarities.

5. Experiments

In our experiments, optimization was performed with AdamW at a learning rate of 3×10^{-5}, accompanied by a linear learning rate scheduler with warm-up. Training used batches of size 16, and input sequences were truncated or padded to a maximum length of 64 tokens. Models were trained for 4 epochs and only saving the model when its L_score improved. helping to prevent overfitting. Dropout regularization with a rate of 0.1 was applied to both hidden and attention layers. All experiments were conducted with a fixed random seed of 42 for reproducibility.

TODO VERIFY THIS I DON'T REALLY UNDERSTAND THE CONFIG @Thösam

5.1. Aggregating LLM Predictions

We conduct a series of experiments to evaluate the impact of aggregating predictions from multiple fine-tuned large language models. Our primary objective is to determine whether ensembling techniques can yield consistent improvements.

We explore the following ensemble strategies:

- **Base vs. Multilingual Variants:** We assess whether combining a base model with its multilingual counterpart—e.g., DistilBERT (base) and DistilBERT (multilingual)—results in better overall performance (see Table 2).
- **Multilingual Ensemble:** We evaluate whether aggregating predictions from multiple multilingual models, such as XLM-RoBERTa and DistilBERT (multilingual), improves classification accuracy across language-diverse inputs (see Table 1).
- **Base Model Ensemble:** We investigate the effectiveness of combining predictions from multiple base models like RoBERTa-base, DeBERTa-base and DistilBERT-base (see Table 1).
- **Base + Large Variants:** We examine whether ensembling base models with their larger counterparts (e.g., RoBERTa-base + RoBERTa-large, DeBERTa-base + DeBERTa-large) leads to enhanced performance due to increased model capacity and diversity (see Table 2).
- **High-Performing + Lightweight Models:** Finally, we test whether adding smaller, faster models to high-performing ensembles yields non-trivial gains, potentially by correcting occasional misclassifications or introducing robustness through architectural diversity (see Table 3).

For each ensemble, we apply both *softmax averaging* and *majority voting* as aggregation strategies. We also report individual model results in terms of the L_score and weighted F1, as well as detailed class-wise performance (see Table 4).

These results highlight the trade-offs between model complexity and performance and showcase which combinations of models and ensembling methods are most effective given our context.

Table 1. Performance comparison of multilingual and base model ensembles on the test set using majority voting (MV) and softmax averaging (SA). Multilingual ensemble: DistilBERT (multilingual) + BERT (multilingual) + RoBERTa (multilingual). Base ensemble: DistilBERT (base, cased) + BERT (base, cased) + RoBERTa (base).

Ensemble Type	Aggregation Method	L_score	Weighted F1
Multilingual	MV	0.8560	0.76
Multilingual	SA	0.8601	0.76
Base	MV	0.8719	0.78
Base	SA	**0.8723**	**0.78**

Table 2. Performance of model ensembles combining base models with multilingual or large counterparts using softmax averaging.

Ensemble Type	Model Pair	L_score	Weighted F1
Base + Multilingual	DistilBERT	0.8524	0.75
Base + Multilingual	BERT	0.8626	0.77
Base + Multilingual	RoBERTa	0.8821	0.80
Base + Large	BERT	0.8738	0.78
Base + Large	RoBERTa	0.8898	0.81
Base + Large	DeBERTa v3	**0.9004**	**0.82**

Table 3. Performance of ensembles combining DeBERTa base (Db), DeBERTa large (Dl), DistilBERT multilingual (Dml), and RoBERTa large (Rl) with majority voting (MV) and softmax averaging (SA).

Model Ensemble	Aggregation	L_score	Weighted F1
Db + Dl + Rl	MV	0.9012	0.82
Db + Dl + Dml	MV	0.9013	0.82
Db + Dl + Dml + Rl	MV	0.9011	0.82
Db + Dl + Rl	SA	0.9020	0.83
Db + Dl + Dml	SA	0.9026	0.83
Db + Dl + Dml + Rl	SA	**0.9034**	0.83

Table 4. Performance comparison across models on the test set. b = base; c = cased; m = multilingual, l = large

Model	L_score	Weighted F1	C / PM / M
DistilBERT (b, c)	0.8478	0.74	7590 / 2132 / 488
DistilBERT (m)	0.8334	0.72	7376 / 2267 / 567
BERT (b, c)	0.8583	0.76	7752 / 2022 / 436
BERT (m)	0.8478	0.73	7454 / 2189 / 567
BERT (l, c)	0.8728	0.78	7975 / 1873 / 362
RoBERTa (b)	0.8822	0.80	8127 / 1760 / 323
XLM-RoBERTa (b)	0.8613	0.77	7840 / 1907 / 463
RoBERTa (l)	0.8869	0.81	8222 / 1667 / 321
DeBERTa v3 (b)	0.8938	0.81	8295 / 1662 / 253
DeBERTa v3 (l)	**0.8975**	**0.82**	8339 / 1648 / 223

5.2. Mitigating Misclassifications with LLM-Generated Variants

To further investigate the limitations of fine-tuned models, we conducted a targeted experiment aimed at correcting the most frequent misclassifications. Specifically, we picked RoBERTa-large and analyzed the misclassified samples in the test set, and explored whether data augmentation through prompting large language models (LLMs) could help address these errors.

TODO: MAYBE JUSTIFY USE OF ROBERTA AND NOT ANOTHER MODEL?

We leveraged LLaMA 3.1 8B Instruct (Grattafiori et al., 2024) to generate two semantically equivalent but lexically and syntactically diverse paraphrases for each misclassified input. These variations were designed to preserve the original meaning while introducing linguistic diversity that might be better captured by the classifier. The model was prompted using a structured instruction template (see Appendix 9.2).

We then evaluated whether aggregating predictions across the original input and its two generated variants—using either *majority voting* or *softmax averaging*—could reduce the error rate. Our results indicate that this strategy leads to a measurable reduction in misclassification errors. While the method is computationally expensive due to the use of a large generative model and the need to perform inference multiple times per sample, it demonstrates the potential of leveraging LLMs for targeted robustness improvements in classification settings.

This experiment suggests that augmentation through high-quality, model-generated variants can act as an effective fallback mechanism, especially when high precision is critical. Future work may explore more efficient variant generation or selective application to optimize the cost-performance trade-off.

TODO SPEAK ABOUT TRANSLATING PHRASE AS WELL?

6. Analysis

Our analysis focuses on understanding the performances between individual models and models with various ensembling strategies.

Among individual transformer models, DeBERTa v3 (large) achieved the highest performance, with an *L_score* of 0.8975 (see Table 4) followed by its base variant and RoBERTa-large. This result is expected, since the size of the model significantly influences the performance.

We also notice that all the multilingual variants have a poorer performance than their corresponding base models. Multilingual models underperformed compared to English-only models when evaluated on English text. This performance gap is likely due to the model's need to support multiple languages, which may come to the detriment of its accuracy in English. Additionally, since the test set contains very little non-English text, there are insufficient samples for these models to take advantage of their multilingual capabilities, resulting in limited performance gains.

As shown in Table 2, combining base and multilingual variants of the same architecture using softmax averaging can help mitigate the performance gap. This approach makes use of the strengths of both models and enhances overall performance by achieving more accurate classification of non-English text in the test set.

Table 1 shows results for aggregating across multiple multilingual and across base models respectively. Base model ensembles using softmax averaging outperformed majority voting, with an *L_score* of 0.8723 and *weighted F1 score* of 0.78. This is expected, as softmax averaging takes into account each model's confidence in its predictions, whereas majority voting simply selects the most frequent label, regardless of how confident each model is. TODO: IS THIS CORRECT ????

Table 2 also evaluates the combination of base and large models. Both RoBERTa and DeBERTa ensembles showed improvements when combining base and large variants, with the DeBERTa (b + l) ensemble achieving the best *L_score* of 0.9004 and *weighted F1 score* of 0.82. Even though, we ensemble models of the same type, they outperform the stand-alone large variant of the corresponding type. The improvement likely comes from the complementary strengths of the base and large models. Although they share the same architecture, differences in model complexity can capture different patterns and may result to diverse and more robust predictions.

Combining multiple models, as shown in Table 3, achieves the highest overall performance. The best result was achieved by using a softmax-averaged ensemble of four models, with a *L_score* of 0.9034 and a *weighted F1 score* of 0.83. This supports the idea that ensembling models of different sizes leverages their complementary strengths, enhancing overall robustness and accuracy.

7. Conclusion

8. Future Work

References

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale, 2020. URL https://arxiv.org/abs/1911.02116.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Go, A., Bhayani, R., and Huang, L. Twitter sentiment classification using distant supervision. In *Proceedings of the Workshop on Language in Social Media (WLSM)*, 2009. URL https://www.scirp.org/reference/referencespapers?referenceid=2790379.

Grattafiori, A. et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

He, P., Gao, J., and Chen, W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2023. URL https://arxiv.org/abs/2111.09543.

Kim, Y. Convolutional neural networks for sentence classification, 2014. URL https://arxiv.org/abs/1408.5882.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

Pang, B., Lee, L., and Vaithyanathan, S. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pp. 79–86. Association for Computational Linguistics, July 2002. doi: 10.3115/1118693.1118704. URL https://aclanthology.org/W02-1011/.

Pennington, J., Socher, R., and Manning, C. GloVe: Global vectors for word representation. In Moschitti, A., Pang, B., and Daelemans, W. (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162/.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., and Bethard, S. (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

9. Appendix

9.1. Paraphrased Input

Table 5. Correcting misclassified examples using two paraphrased versions of each input. Final predictions are obtained using either majority or averaged softmax scores. N = Negative, Ne = Neutral, P = Positive.

True/Pred	**Orig.**	**Aggregation Method**	
		Maj. Vote (N/Ne/P)	**Softmax Avg (N/Ne/P)**
P/N	129	90 / 1 / 38	81 / 4 / 44
N/P	192	57 / 7 / 128	68 / 10 / 114
Ne/N	358	273 / 55 / 30	255 / 69 / 34
Ne/P	473	13 / 54 / 406	21 / 68 / 384
P/Ne	463	11 / 314 / 138	14 / 299 / 150
N/Ne	373	110 / 243 / 20	116 / 232 / 25

9.2. Standard machine learning approach

Table 6. Performance comparison across models on the test set. N = Negative, Ne = Neutral, P = Positive.

Model	**L_score**	**Accuracy**	**Precision (N/Ne/P)**	**Recall (N/Ne/P)**	**F1-score (N/Ne/P)**	**Weighted F1**
LogReg	0.8041	0.67	0.61 / 0.68 / 0.69	0.39 / 0.86 / 0.57	0.48 / 0.76 / 0.62	0.66
LogReg + expand contractions	0.8057	0.67	0.62 / 0.68 / 0.70	0.40 / 0.86 / 0.56	0.49 / 0.76 / 0.62	0.66
LogReg + remove emails	0.8044	0.67	0.61 / 0.68 / 0.70	0.39 / 0.86 / 0.57	0.48 / 0.76 / 0.62	0.66
LogReg + remove urls	0.8042	0.67	0.61 / 0.68 / 0.69	0.39 / 0.86 / 0.57	0.48 / 0.76 / 0.62	0.66
LogReg combined	0.8063	0.67	0.62 / 0.68 / 0.70	0.40 / 0.86 / 0.57	0.49 / 0.76 / 0.62	0.66
RF	0.7924	0.64	0.61 / 0.63 / 0.68	0.28 / 0.89 / 0.49	0.39 / 0.73 / 0.57	0.61
XGBoost	0.8028	0.65	0.67 / 0.62 / 0.74	0.29 / 0.92 / 0.47	0.41 / 0.74 / 0.57	0.62
MLP	0.7782	0.64	0.53 / 0.69 / 0.61	0.40 / 0.77 / 0.60	0.46 / 0.73 / 0.60	0.63
RoBERTa	0.8869	0.81	0.77 / 0.83 / 0.79	0.74 / 0.83 / 0.81	0.76 / 0.83 / 0.80	0.81

9.3. LLM Prompts

TODO: PICK ONE OUT OF THE TWO DEPENDING ON TRANSLATION OR NOT

Prompt for generating paraphrases and translating sentences if necessary:

```
You are assisting with a sentiment classification task.

Given the input sentence, first output it as-is or translate it to English if necessary.
Then, generate exactly 2 paraphrased versions in English that preserve both the sentiment
and meaning, but use different words or sentence structures.

Output format (exactly 3 lines):
- Start with "<ORIGINAL> " followed by the original or translated sentence.
- Then, on two separate lines, start each with "<VARIATION> " followed by a paraphrased
version.
- Maintain this exact format, no extra text or lines.

Input: <<INPUT>>
```

Prompt for generating paraphrases:

```
You are helping with a sentiment classification task.

Given the following input sentence, generate exactly 2 paraphrased versions that
express the same sentiment and meaning, but use different words or sentence structure.

Important rule:
- Always start each line with "<VARIATION> " followed by the paraphrased sentence.
- Return only the 2 lines.

Text: "<<INPUT>>"
```