Computational Intelligence Lab: Sentiment Analysis

Kai Lascheit, Tjark Behrens, Jacob Hunecke
ETH Zürich
klascheit@student.ethz.ch
tbehrens@student.ethz.ch
jhunecke@student.ethz.ch

Abstract

Sentiment analysis is a vital component of natural language understanding, applied across various domains including customer service, healthcare, and business intelligence. Recent advancements in transformer-based language models have provided a foundation to further improve the accuracy and reliability of current approaches, which we aim to exploit in this paper. We present an ensemble of three state-of-the-art transformers, fine-tuned on task-specific data and combined using aggregation layers. Our method has been tested on a Twitter dataset containing two million samples, achieving an accuracy of over 90%. We compare this approach to two baselines: a random forest classifier and a Recurrent Neural Network method, which achieved 78% and 83% accuracy respectively. The results demonstrate the superior accuracy of our ensemble method. Source code with a detailed explanation on how to reproduce the results mentioned in this paper, can be found in our GitHub repository (https://github.com/jlohunecke/CIL_sentiment_classification).

1. Introduction

In the era of big data, modern companies and institutions face the challenge of processing large amounts of text to extract valuable information such as customer satisfaction and users' personal preferences. This data can be used to improve products and services, identify trends in consumer behavior, and personalize experiences. Social networks are a major provider of user-centered data, recording hundreds of millions of new posts each day. However, this data is particularly challenging to analyze as it consists largely of unstructured text that loosely follows grammatical rules and uses special symbols such as hashtags and emojis.

Conventional sentiment analysis methods, like the Random Forest classifier, require extensive preprocessing and struggle to capture the complexity of unstructured data. Alternative approaches employ Recurrent Neural Networks (RNNs), which require less preprocessing and show advancements in accuracy. However, we believe that performance can be improved further by exploiting the recent advances in transformer architectures.

In this paper, we present an ensemble of three state-of-the-art transformers. We then combine the latent representations of these models using aggregation layers before the output layer, creating an ensemble classifier. We evaluated the performance of our proposed method on a Twitter dataset containing two million posts, comparing the measured accuracy to two baselines: one using a Random Forest classifier and one using an RNN.

2. Related Work

Traditionally, sentiment analysis relied on classical machine learning techniques such as Bag-of-Words [12] representations and N-grams [3], combined with classification algorithms like Support Vector Machines (SVMs) [5] and Random Forests [2]. Examples of such methods are presented in [10] and [1]. The latter paper reports an accuracy of 75% using a Random Forest approach on a Twitter dataset. We use a similar method as one of our baseline classifiers. The notable inaccuracy of traditional approaches, as exemplified by this study, has led researchers to explore neural network architectures, specifically Recurrent Neural Networks (RNNs).

A recent study tested a RNN-based method on a Twitter dataset and reported an accuracy of 80% [13], marking a significant improvement over the Random Forest approach. Given the popularity and improved performance of RNNs for sentiment analysis, we select an RNN approach as our second baseline. However, the misclassification of 20% of Tweets indicates that there is still considerable room for improvement. To address these limitations, we turn to transformer-based architectures, specifically BERT [4], RoBERTa [8], and BART [7]. BERT, the oldest of these approaches, utilizes bidirectional context and is trained using Masked Language Modeling and Next Sentence Predic-

tion. RoBERTa is a robustified version of BERT, enhancing performance by optimizing the training process. BART combines a bidirectional encoder, like BERT, with an autoregressive decoder, similar to GPT [14].

We empirically select these three architectures for our study, as explained in the experiments section [4]. To leverage the strengths of all three transformers, we create an ensemble by adding aggregation layers for output computation.

3. Method

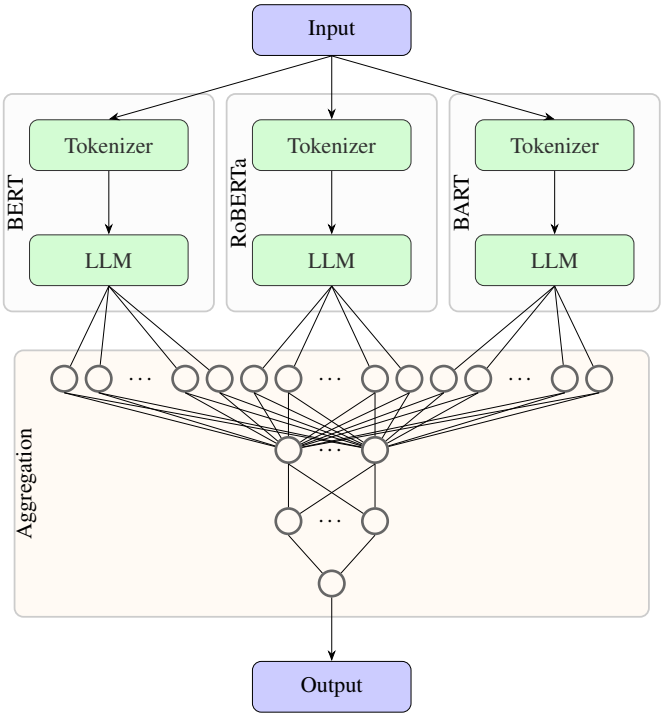

Figure 1. Stacking-based model architecture consisting of 3 parallel pre-trained LLM branches and a succeeding meta model computing the binary classification output through a FCNN

As shown in 1, the architecture follows a stacking design with three parallel LLM branches and a downstream aggregation module that acts as a metamodel. The three LLM branches compute latent representations of the tweets, which are then passed to the aggregation module, which determines the final classification output.

3.1. Branches

The three branches are based on three different pre-trained LLM architectures, including a model-specific tokenizer and a pre-trained LLM model. To obtain a latent representation from the LLMs, the final output layer of the LLM models is removed. The representations returned by the LLMs are then used as the latent representation.

3.1.1 Branch 1 - BERT

The first branch uses the BERT model, which was developed by researchers at Google and launched in 2018. It is based on a bidirectional transformer architecture and has been pre-trained on a corpus of 3.3 billion words. BERT performs particularly well at understanding context within sentences due to its bidirectionality [4].

3.1.2 Branch 2 - RoBERTa

The second branch is based on the RoBERTa model [8], which is an extended version of BERT with respect to hyperparameter choices. In contrast to the original BERT model, it was trained on a larger training data set for a longer training time. This leads to a more robust model which has demonstrated superior performance on various benchmarks compared to BERT as detailed in [8].

3.1.3 Branch 3 - BART

The last branch uses the pre-trained BART model [7], which was developed by Facebook and released in 2019. BART is based on a transformer-based encoder-decoder architecture, so unlike the previous two models, it can not only understand text, but also generate it. Furthermore, it is particularly strong in sequence-to-sequence tasks, which can enhance the classification task by better understanding tweet sequences and their transformations.

3.2. Aggregation Module

The aggregation module acts as a metamodel, concatenating each of the 1024-dimensional latent representations of the three branches into a final classification output. It consists of two fully connected hidden layers with 512 neurons each and a final output layer that returns the binary classification output.

3.3. Motivation

We decide to employ an ensemble of BERT, RoBERTa, and BART as this allows us to combine the individual strengths of each model and reduce the risk of overfitting. BERT's bidirectional transformer architecture excels in contextual understanding within sentences, while RoBERTa's extensive training on larger datasets offers enhanced robustness and superior performance. BART's encoder-decoder structure is particularly strong in sequence-to-sequence tasks, which aids in comprehending and transforming tweet sequences. The ensemble leverages these complementary capabilities to capture a broader range of linguistic patterns, leading to more accurate and reliable classifications. Furthermore, this approach addresses the issue of individual model biases, resulting in improved generalisation and robustness, as demonstrated in the experimental results presented in section [4].

4. Experiments and Results

To evaluate our model, it was benchmarked against two baseline models and several modified versions of the parallel multi-branch architecture. As an evaluation metric, we consistently examined validation loss and validation accuracy (see equation 1) across all of our experiments.

$$\text{Val. Acc.} = \frac{\sum_{(X_i,Y_i)\in D_{Val}} \mathbb{1}[Y_i = f(X_i)]}{|D_{Val}|} \quad (1)$$

All the models were trained on the Twitter data set (80% training and 20% validation data) using the binary cross entropy loss.

4.1. Dataset

The Twitter dataset consists of 2 million tweets in total - 1 million per class. The dataset was generated by taking tweets containing positive or negative emojis, labeling the tweets as positive or negative accordingly, and finally removing the emojis.

4.2. Baselines

As baselines, we showcase both a classical ML-approach and an RNN to highlight two traditional approaches to sentiment classification and compare them to our transformer-based architecture. Specifically, we employ a Random Forest classifier [2], due to its comparably high effectiveness and robustness among classical approaches, and a bidirectional GRU-based RNN which is particularly suited for sentiment classification as it is designed to understand sequences and contexts of words [13]. Inputs to these two classifiers are preprocessed by removing stop words and punctuation and applying word stemming, lemmatization and embedding. The final component is of particular importance for the model performance and is therefore included in our experimental analysis. We compare two embedding techniques: TF-IDF and GloVe [11]. While TF-IDF offers a simple and easily interpretable embedding, GloVe provides a more complex approach, capable of capturing contexts and relationships between words. As our experiments will show, GloVe's contextual embeddings significantly benefit the performance of RNNs by enhancing the inputs with additional semantic information, while TF-IDF is more effective when combined with the Random Forest classifier, as this model is designed to handle the high-dimensional sparse data that TF-IDF provides.

4.3. Experiments

During the development of the multi-branch model, we conducted experiments with several versions of our proposed architecture. These included varying the number of branches and the type of models used in those branches. We also explored different aggregation strategies such as soft averaging, majority voting, and stacking using a meta-model [9]. Finally, we tested two different transfer learning approaches - in the first we trained all layers of the model from the beginning (classical), while in the second we followed a two-stage training approach, freezing the weights of the first x layers of the pre-trained LLMs for 50% of the training epochs and training the whole model for the remaining 50% of the epochs. An overview of the experiments conducted, including model specifications and results, can be found in table 1. The final model weights are chosen to minimize the validation accuracy. We also conducted experiments using Stochastic Weight Averaging [6] as a generalization method, however, with our experimentally chosen hyperparameter configuration the minimum validation accuracy approach turned out to lead to more accurate results.
Due to computational limitations, the benchmarking in section 4.4 was performed on the small dataset containing only 10% (200k tweets) of the total training data. However, the qualitative analysis in section 4.5 is based on results ob-

Model	No. of Branches	LLM Models	Agg. Strategy	Training Approach	Val. Loss	Val. Acc.
Random Forest Classifier (TFIdf)	-	-	-	-	-	78.14%
GRU (TFIdf)	-	-	-	-	0.4413	77.16%
Random Forest Classifier (Glove)	-	-	-	-	-	70.68%
GRU (Glove)	-	-	-	-	0.4184	83.59%
Single-branch model	1	BERT	-	classical	0.3053	88.12%
Single-branch model	1	RoBERTa	-	classical	0.2719	89.02%
Single-branch model	1	BART	-	classical	0.2829	89.15%
Single-branch model	1	BERT	-	2-stage	0.2801	88.21%
Single-branch model	1	RoBERTa	-	2-stage	0.2545	89.43%
Single-branch model	1	BART	-	2-stage	0.2643	89.07%
Multi-branch model	3	BERT, RoBERTa, BART	Majority Vote	2-stage	0.2523	89.76%
Multi-branch model	3	BERT, RoBERTa, BART	Soft-averaging	2-stage	0.2511	89.82%
Multi-branch model	3	BERT, RoBERTa, BART	Stacking	2-stage	**0.2502**	**89.91%**

Table 1. Comparison of experimental results (based on reduced dataset, containing 10% of the data from the full dataset) between baseline models and different configurations of the parallel multi-branch architecture based on validation loss and accuracy

tained by training the final model on the full dataset (2mn tweets).

4.4. Results

The experimental results, summarized in Table 1, demonstrate a clear performance distinction between baseline models and various configurations of the parallel multi-branch architecture. Especially for the larger LLM architectures, overfitting was quite apparent in the training routine. Hence, the selection of the models for evaluation and comparison is solely based on the best achieved validation performance.

Among the baseline models, the GRU implementation using Glove embeddings achieved the highest validation accuracy at 83.59%. In comparison, the single-branch models using large language models (LLMs) like BERT, RoBERTa, and BART showed an improved performance, with RoBERTa achieving around 89% accuracy. Moreover, one can observe that the results of the experiments using the single-branch models are slightly improved when using the 2-stage training approach, compared to the classical one. We identified that keeping the LLM-backbone freezed during initial training helps regularize the attached aggregator. The reason for this is that a completely unfrozen architecture tends to overfit fast (after 2-3 epochs), leading to an uncalibrated aggregation module. These different training setups are visualized in figure 2 for our RoBERTa single-branch model. While the unfrozen model overfits already after the second epoch (blue), we employ a more steady training regime by freezing the LLM architecture for the first half of the epochs (orange).

Finally, we found however that the multi-branch models (3 branches) significantly outperformed the single-branch models. Our soft-averaging and majority vote approaches [9] are built upon the single-branch, two-stage LLM models. This setup does not require any re-training, thus, there is no data regarding the validation loss. These two ensembling methods already show the superiority of this idea. However, going one step further, the model using BERT, RoBERTa, and BART with a stacking aggregation strategy (our final model) attained the highest validation accuracy of 89.91%. Training all three branches simultaneously end-to-end offered the most representational power of the three concatenated hidden representations while allowing the fully-connected aggregation module to identify the most important parts for sentiment analysis.

4.5. Analysis of Final Model Results

The final model (multi-branch architecture using BERT, RoBERTa, and BART with stacking aggregation) trained on the full dataset achieved a validation accuracy of 90.41%. We qualitatively evaluated the results of our best-performing model on the validation data. By looking at

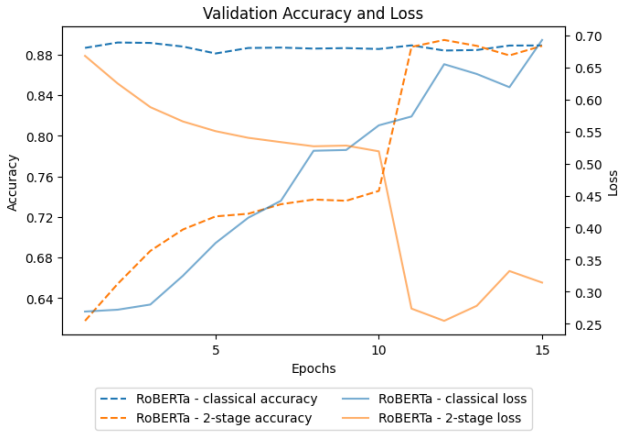

Figure 2. Comparison of validation accuracy and loss between the classical and the 2-stage training strategy for the single-branch RoBERTa model

Table 2 one can see that the model predicts slightly more false positives than false negatives. For the task of sentiment analysis, this can be interpreted as a tendency of the model towards positive emotion instead of negative. More mistakes are made for tweets that are actually negative, but are misclassified as positive sentiment. However, the confusion matrix emphasizes that our model is generally well-calibrated.

Table 2. Confusion matrix containing absolute numbers of true positives/negatives and false positives/negatives obtained on the validation data set using our final model

		Predicted	
		Positive	**Negative**
Actual	**Positive**	229025	20975
	Negative	26410	223590

The model confirms its strong performance by achieving 90.32% on the public part of the test set of the Kaggle competition.

5. Conclusion

The above results highlight the effectiveness of multi-branch architectures and advanced aggregation strategies in improving model accuracy for Twitter sentiment analysis. It is likely that the performance of the architecture can be further improved by integrating more branches into the model using more distinct pre-trained LLMs. In addition, further hyperparameter tuning regarding the layer width and depth of the stacking module could lead to even better classification accuracies.

References

[1] Bahrawi Bahrawi. Sentiment analysis using random forest algorithm-online social media based. *Journal of Information Technology and Its Utilization*, 2:29, 12 2019. 1

[2] L Breiman. Random forests. *Machine Learning*, 45:5–32, 10 2001. 1, 3

[3] William Cavnar and John Trenkle. N-gram-based text categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, 05 2001. 1

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 10 2018. 1, 2

[5] Theodoros Evgeniou and Massimiliano Pontil. Support vector machines: Theory and applications. volume 2049, pages 249–257, 09 2001. 1

[6] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 3

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. pages 7871–7880, 01 2020. 1, 2

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 07 2019. 1, 2

[9] Ammar Mohammed and Rania Kora. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*, 35(2):757–774, 2023. 3, 4

[10] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. pages 412–418, 01 2004. 1

[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 3

[12] Wisam Qader, Musa M. Ameen, and Bilal Ahmed. An overview of bag of words;importance, implementation, applications, and challenges. pages 200–204, 06 2019. 1

[13] Merin Thomas and Dr C A. Sentimental analysis using recurrent neural network. *International Journal of Engineering and Technology(UAE)*, 7:88–92, 08 2018. 1, 3

[14] Gokul Yenduri, Ramalingam Murugan, Chemmalar Govardanan, Y Supriya, Gautam Srivastava, Praveen Reddy, Deepti Raj, Rutvij Jhaveri, Prabadevi B, Weizheng Wang, Athanasios Vasilakos, and Thippa Gadekallu. Gpt (generative pre-trained transformer)— a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*, PP:1–1, 01 2024. 2