Impact of Preprocessing and Model Ensemble on Twitter Sentiment Classification

Dominic Steiner   Yiming Wang   Yufei Zhang   Xin Hong
Group: Cutie Pies, Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—This paper explores the ensemble of various transformer-based models and preprocessing techniques to enhance sentiment classification accuracy on Twitter data. An examination of different preprocessing methods reveals that techniques such as misspelling corrections, hashtag splitting, and emoticon replacement can significantly improve model performance. Unexpectedly, traditional techniques such as the removal of stopwords may have adverse effects. Among the tested models, the Twitter RoBERTa model, combined with the aforementioned preprocessing methods, yielded the highest accuracy. Nevertheless, the combination of our top-performing models in an ensemble achieved even more accurate results.**

I. Introduction

In the age of digital communication, microblogging and text messaging have seen significant growth over the last decade. This proliferation of information has increasingly amplified the need for automatic methods to understand, interpret, and infer the sentiments expressed in text data. It would uncover unknown insights of public opinion, trends, and perceptions.

Sentiment classification of text leverages natural language processing (NLP) and machine learning (ML) to determine whether a text expresses a positive or negative sentiment. However, directly applying pretrained ML models often fails to yield optimal results due to the unique and distinct characteristics of the data. Tweets, for instance, are known for their short length and frequent use of hashtags, slang, abbreviations, emoticons, etc. Therefore, preprocessing is essential to handle these features. Selecting suitable models and fine-tuning their parameters are also critical to achieve a high classification accuracy. This research aims to classify sentiments in tweets. The training data for the ML models consists of a dataset comprising 2.5 million pre-classified (either positive or negative) tweets.

This research offers significant insights into sentiment analysis for social media texts, presenting a systematic approach to classifying tweet sentiment.

II. Models and Methods

To tackle the challenge of sentiment analysis on tweets, we adopted a variety of methods and models, drawing inspiration from existing studies in the field. Besides choosing and ensembling models, a significant portion of our work was dedicated to executing several preprocessing steps and validating their effectiveness. We also devoted substantial resources to appropriate data partitioning and precise fine-tuning of model parameters.

A. Baseline Models

We set up some baseline models as benchmarks to test the performance of more complex models.

1) Logistic Regression [1]: In the early stages of this research, we employed a simple logistic regression model using a bag-of-words representation of the tweets. This model demonstrated the classification power of a linear model and served as a baseline for more complex, transformer-based models.

2) LSTM [2]: Like before, we utilized the bag-of-words representation of the tweets. Subsequently, a Long Short-Term Memory (LSTM) network was employed to process the sequential nature of the text and capture dependencies over different time steps.

3) GloVe [3] + CNN [4]: We averaged the pre-trained Global Vectors for Word Representation (GloVe) word embeddings to create representations for tweets. Then, we employed a 1D Convolutional Neural Network (CNN) to capture local patterns and features, followed by a ReLU layer and a max pooling layer. After that, a fully connected layer and a softmax layer were used to do the classification.

4) BERT-mini [5]: Bidirectional Encoder Representations from Transformers (BERT) [6] is a revolutionary transformer-based language model that adopts a masked language modeling (MLM) objective during pre-training. Its key innovation lies in bidirectional context, allowing it to capture deep contextual relationships in text by considering the entire input sequence in two directions.

BERT-mini is a smaller variant of BERT, with 4 layers and 256 hidden units. We chose it as a baseline for transformer-based models due to its computational efficiency. We averaged the word embeddings generated by BERT-mini to get the sentence embeddings, and utilized a fully connected layer with a sigmoid activation to perform the classification. The BERT-mini model was fine-tuned for our specific task during the training process, using the Adam optimizer [7] and the binary cross entropy loss function.

B. Transformers

We proceeded with more complex transformer-based models, experimented with different tweet embedding generation methods, and fine-tuned the models.

1) RoBERTa [8]: The Robustly Optimized BERT Pretraining Approach (RoBERTa) is a variant of BERT. RoBERTa differs in its training approach, particularly by dynamically adjusting the masking pattern applied to the training data and training with much larger mini-batches and learning rates, as well as longer training duration.

Typically, the final embedding of a tweet was generated by pooling the outputs from the last hidden layers of the transformer-based models. However, layers in BERT and similar transformer-based models are designed to learn hierarchical representations of the input text, thus containing different levels of information. The bottom layers capture the surface features of the text, the intermediate layers extract syntactic features, and the top layers catch semantic features [9]. Therefore, we tried several methods to include some intermediate layers in generating tweet embeddings as they might add more contextual information to the sentence representation.

We also tried to use the weighted average of the last four hidden layers to create a different kind of sentence embedding. We generated the second type of sentence embedding by feeding the last four hidden layers into a CNN. The filters moved along the sequential direction of the tweet sentence with the size of three. Additionally, we repeated this procedure on the intermediate four hidden layers. The results of the embeddings from the last four hidden layers and the middle four hidden layers were concatenated to form the fourth type of sentence embedding. To perform the classification task, we applied a fully connected layer followed by a sigmoid layer on the generated sentence embeddings. We evaluated these four embedding methods on RoBERTa, to decide which one to use in subsequent experiments.

2) XLNet [10]: It is another transformer-based language model which differs from BERT in its training strategy. XLNet's strategy is based on permutations rather than masked language models. This difference allows XLNet to capture dependencies between all words in a sentence, unlike BERT. We extended the XLNet model in the same manner as described in the previous section.

3) Twitter RoBERTa [11][12]: This model is a sentiment-specific version of RoBERTa and pre-trained using a Twitter dataset. It is optimized for Twitter sentiment analysis tasks. The model classifies each tweet as either positive, neutral, or negative sentiment. We enhanced the model by incorporating an additional dense layer with a sigmoid activation function to reduce the output to a single value for binary classification. We fine-tuned the model for our specific task, using the Adam optimizer with binary cross-entropy loss.

4) Twitter XLM RoBERTa [13]: This model is different from Twitter RoBERTa; it is a large-scale multilingual pre-trained language model trained on various languages. It is based on RoBERTa and improved to handle tweet sentiment analysis. We fine-tuned this model specifically for our task, using the same network architecture and hyperparameters as in the fine-tuning process of the original RoBERTa model.

5) RoBERTa TweetEN [14]: Like the two previous models, it is also a derivative of RoBERTa. The Cardiff NLP group trained, validated, and tested the model on a balanced multilingual tweet sentiment dataset with tweets in eight languages. While this model and Twitter XLM RoBERTa share a common base in the RoBERTa architecture, it distinguishes itself with dedicated fine-tuning on multilingual tweet sentiment. For our study, we employed this model focusing on English tweets. In line with the original RoBERTa model, we utilized the same network architecture and hyperparameters in the fine-tuning process.

6) Emoji RoBERTa [15]: This model is trained on the TweetEval dataset [16] and designed specifically to predict the most suitable emoji for a given input. Despite initially seeming like an unconventional choice, we were interested in trying this model because our dataset is labeled based on the emojis included by the posters in their tweets. By leveraging this model's ability to predict the correct emojis, we might potentially enhance our sentiment predictions, particularly for tweets containing irony.

The model outputs predicted probabilities for 20 different emojis. To adapt it for sentiment classification, we introduced two additional dense layers at the end of the model. One layer utilizes the ReLU activation function, while the other uses sigmoid activation. The final layer reduces the 20 activations to a single value, enabling predictions for either positive or negative sentiment.

C. Preprocessing

In this section, we conducted two primary preprocessing steps: first, data cleaning involving various preprocessing techniques for content manipulation, and second, data splitting to standardize tweet lengths as inputs for our models.

1) Data Cleaning: Tweets deviate significantly from conventional written texts, exhibiting characteristics such as hashtags, abbreviations, slang, misspellings, among others, even after tokenization in the provided training data. To address these challenges, we conducted experiments with several preprocessing techniques.

Hashtag Splitting: We utilized the hashtags in the original tweets to enhance our model. The dataset contained 364,709 hashtags, averaging at 0.14 hashtags per tweet. 77% of all unique hashtags occurred only once, and 97% occurred 10 times or fewer. Most hashtags consisted of concatenated common words with a ”#” prefix, such as *#sometimesijustwant*, *#notcool*, and *#shutupandkissme*. We hypothesized that breaking these hashtags down into constituent words using the pywordsegment Python package [17] could reveal valuable information.

Emoticon Replacement: Emoticons were initially used to label our tweet dataset, but some traditional ASCII emoticons remain. To address this, we replaced all emoticons with over 100 occurrences with adjectives that expressed the sentiment they conveyed, thus enabling an NLP model to train on a richer vocabulary and better understand the underlying meaning of these emoticons.

Model	Preprocessing Strategy	Valid. Acc. %
Logistic Regression	None	80.3
	No punctuation	80.3
	Spellchecking	80.2
	Hashtag splitting	80.4
	Emoticon replacement	80.3
	Stopwords removal	79.9
	Hashtag + Emoticon + spellcheck	80.3
BERT-mini	None	87.1
	No punctuation	86.5
	Spellchecking	88.1
	Hashtag splitting	87.8
	Emoticon replacement	87.9
	Stopwords removal	86.5
	Hashtag + Emoticon + spellcheck	87.8

Table I
EVALUATION OF PREPROCESSING STRATEGIES

Spellchecking: Tweets often contain spelling mistakes, typical of rapid online communication. Consequently, we utilized the autocorrect Python package [18] to correct spellings. Our hypothesis was that by reducing the frequency of misspelled words, we could enhance the model's performance, since the correct and misspelled tokens would no longer be treated as separate entities.

Removing Punctuations: We observed that the dataset contained tokens consisting solely of single punctuation or other symbol characters. We conducted experiments to remove these tokens, as they typically lack semantic meaning and their removal could potentially enhance the model's performance.

Stopwords: Stopwords, which include articles, conjunctions, and prepositions, are high-frequency but low-semantic-value terms. We hypothesized that removing these stopwords could benefit certain models by allowing them to focus on more meaningful words.

We empirically evaluated the impact of preprocessing methods on two model architectures: linear regression and BERT-mini. For the former, we utilized a bag of words vectorizer, capped at a maximum of 5000 features, with a regularization value of $C = 10^{-5}$, and we halted training after 100 iterations. In contrast, the latter was trained using a learning rate of 10^{-5} and a batch size of 16. After two epochs of training, we determined the validation accuracy for each preprocessing technique.

Our specific preprocessing experiment findings are detailed in Table I. For the linear regression model, the impact of preprocessing strategies was minimal, with hashtag splitting offering a slight accuracy boost and stopwords removal and spellchecking leading to a marginal decline. However, in the case of the BERT-mini model, hashtag splitting, emoticon replacement, and spellcheck modestly improved accuracy. As a result, we adopted these three techniques for future transformer models.

2) Data Splitting: Upon analyzing the distribution of token counts in tweets, as illustrated in Figure 1, an intriguing observation emerged: 99.99% of all tweets contained 45 tokens or fewer. Based on this finding, we determined that setting the input tweet length for our models at 45 would be appropriate. Shorter tweets were handled with padding, while longer ones were truncated.

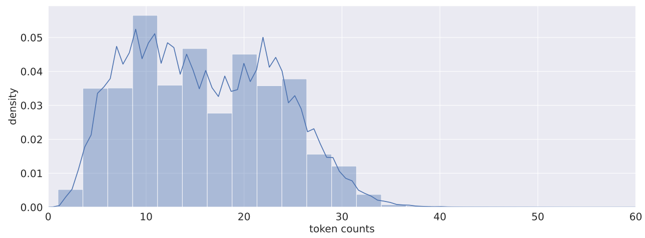

Figure 1. Distribution of token counts in tweets

Model	Training Acc. %	Validation Acc. %
Linear Regression	80.5	80.3
LSTM	85.7	85.7
GloVe (Trainable=False) + 1DCNN	83.8	81.6
GloVe (Trainable=True) + 1DCNN	88.1	85.2
BERT-mini	89.0	87.5

Table II
ACCURACY OF BASELINE MODELS WITHOUT PREPROCESSING

D. Ensemble

Ensemble Learning can typically improve overall performance by combining the predictions of multiple models. We chose the top 3 models that gave highest validation accuracy and evaluated their ensemble result in a hard voting setting.

E. Model Parameters

We began our experiment by setting the learning rate at 10^{-5}, adhering to the standard Adam learning rate, and assessed different batch sizes including 16, 32, and 64. We discovered that although the performance remained consistent across these sizes, batch size 32 was able to yield results most efficiently. Thus, we decided to retest all models using a standard learning rate of 10^{-5} and a batch size of 32 for a uniform comparison. We reported the maximum validation accuracy achieved by each model. We trained each model until the validation accuracy ceased to increase, typically within 4 epochs.

III. RESULTS

We initiated the project by evaluating simpler models. Table II displays the results achieved by these baseline models. Even the relatively simple linear regression model managed to achieve a relatively good accuracy of 80.3%. Overall, the BERT-mini model achieved the highest accuracy without any sign of overfitting. This outcome provided us with the confidence to move on to larger transformer models.

Table III indicates that neither the weighted averaging of the last four hidden layers nor the utilization of intermediate

Model	Representation (Hidden layers used)	Training Acc. %	Validation Acc. %
XLNet + dense	last one	93.6	90.0
XLNet + dense	weighted avg. last four	93.4	90.0
XLNet + CNN	last four	93.6	90.1
XLNet + CNN	last four + mid four	93.5	90.0
RoBERTa + dense	last one	93.5	90.3
RoBERTa + dense	weighted avg. last four	93.6	90.3
RoBERTa + CNN	last four	93.6	90.2
RoBERTa + CNN	last four + mid four	94.2	90.2

Table III
ACCURACY OF MODELS USING DIFFERENT TWEET PRESENTATION METHODS

Model	Training Acc. %	Validation Acc. %
RoBERTa TweetEN	93.4	90.3
Twitter XLM RoBERTa	93.6	90.4
Twitter RoBERTa	92.5	90.8
Emoji RoBERTa	91.3	90.2

Table IV
ACCURACY OF SPECIALIZED PRE-TRAINED MODELS

hidden layers with CNN enhances the performance, when compared to using only the last hidden layer. This holds for both XLNet and RoBERTa. In fact, incorporating CNN in representation generation and classification might slightly degrade performance. Therefore, considering computational efficiency, we chose to use only the last hidden layer to generate tweet classifications in subsequent experiments.

Until now, we have used transformer models trained for general NLP tasks. However, in Table IV, we present the results for pre-trained models specifically trained on Twitter datasets. Notably, the Twitter RoBERTa model exhibited the most impressive performance. We noted these favorable results when we applied the emoticon replacement, hashtag splitting, and spellchecking preprocessing techniques, with a learning rate of 10^{-5} and a batch size of 32 over 3 epochs.

With this ensemble approach that combines predictions from the RoBERTa TweetEN, Twitter XLM RoBERTa, and Twitter RoBERTa models, we significantly improved our final validation accuracy, achieving 91.1% up from 90.8%. This increase is quite substantial given the inherent difficulty of advancing performance beyond the 90% accuracy threshold. Additionally, our ensemble model performed quite well on the Kaggle test dataset, achieving a score of 91.0%, thereby outperforming all individual models in the experiment.

IV. DISCUSSION

Contrary to expectations, standard data cleaning techniques like punctuation removal and stopword deletion, exhibited a surprising negative effect on transformer-based models during our experiment (Table I). This could be attributed to the fact that transformer-based models depend significantly on context, especially when compared to other pre-trained embedding models like GloVe. Given the short length of tweets, the preservation of complete context is crucial when fine-tuning transformer-based models and generating accurate embeddings. Conversely, strategies such as splitting hashtags into individual words and replacing emoticons with adjectives can be seen as techniques to restore more complete and accurate context, thereby potentially improving the model's performance.

Our results demonstrate that incorporating information from additional hidden layers did not enhance classification accuracy. We hypothesize that while different hidden layers are expected to capture distinct contextual information, the limited length of tweets makes this context generation harder. Consequently, merging information from multiple hidden layers may lead to redundancy rather than performance enhancement.

Moreover, the use of CNNs for text classification did not result in improved accuracy. One of the main reasons for deploying CNNs in text classification is their capacity to detect local patterns through sliding windows, which hold significant semantic or syntactic information. However, embeddings generated by transformer-based models already comprise comprehensive contextual information, extracted from their powerful hidden layers. Hence, using a fully connected layer for classification is not inferior to CNNs.

Future research should focus on improving preprocessing techniques tailored for social media data, such as refining hashtag splitting methods to address errors like "*#justinbieber*" split into "*just in bieber*". Studies could also explore various embedding techniques, including alternative strategies for combining hidden layer information or employing layers like recurrent neural networks (RNNs) [19]. Moreover, customizing transformer models to suit specific social media platforms, akin to the Twitter RoBERTa model, could lead to more precise sentiment analysis, generating more valuable insights into social media sentiment trends.

V. SUMMARY

This research emphasizes the significance of carefully selecting preprocessing methods and smartly ensembling models based on data characteristics. The model ensemble of RoBERTa TweetEN, Twitter XLM RoBERTa, and Twitter RoBERT, combined with specific preprocessing methods, outperformed all the individual models tested in this research scope. Remarkably, conventional data cleaning techniques, such as removing punctuation and stopwords, had adverse effects on transformer-based models.

These findings highlight the necessity to customize preprocessing steps in accordance with the unique characteristics of social media data and offer valuable insights for future research regarding model selection, ensemble strategies, and fine-tuning methods.

References

[1] S. P. Morgan and J. D. Teachman, "Logistic regression: Description, examples, and comparisons," *Journal of Marriage and Family*, vol. 50, no. 4, pp. 929–936, 1988, accessed: 31-07-2023. [Online]. Available: https://doi.org/10.2307/352104

[2] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[3] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[4] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[5] P. Bhargava, A. Drozd, and A. Rogers, "Generalization in nli: Ways (not) to go beyond simple heuristics," 2021.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[9] G. Jawahar, B. Sagot, and D. Seddah, "What does bert learn about the structure of language?" in *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[10] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[11] J. Camacho-collados, K. Rezaee, T. Riahi, A. Ushio, D. Loureiro, D. Antypas, J. Boisson, L. Espinosa Anke, F. Liu, E. Martínez Cámara *et al.*, "TweetNLP: Cutting-edge natural language processing for social media," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Abu Dhabi, UAE: Association for Computational Linguistics, Dec. 2022, pp. 38–49. [Online]. Available: https://aclanthology.org/2022.emnlp-demos.5

[12] D. Loureiro, F. Barbieri, L. Neves, L. Espinosa Anke, and J. Camacho-collados, "TimeLMs: Diachronic language models from Twitter," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 251–260. [Online]. Available: https://aclanthology.org/2022.acl-demo.25

[13] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[14] F. Barbieri, L. Espinosa Anke, and J. Camacho-Collados, "XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, Jun. 2022, pp. 258–266. [Online]. Available: https://aclanthology.org/2022.lrec-1.27

[15] F. Barbieri, J. Camacho-Collados, F. Ronzano, L. Espinosa-Anke, M. Ballesteros, V. Basile, V. Patti, and H. Saggion, "Semeval 2018 task 2: Multilingual emoji prediction," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 24–33.

[16] F. Barbieri, J. Camacho-Collados, L. Espinosa-Anke, and L. Neves, "TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification," in *Proceedings of Findings of EMNLP*, 2020.

[17] G. B. David, "Pywordsegment," https://pypi.org/project/pywordsegment/, 2021.

[18] F. Pires, "Spelling corrector in python," https://github.com/filyp/autocorrect, 2021.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Institute for Cognitive Science, University of California, San Diego, California, Tech. Rep. ICS 8504, Sept 1985.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Impact of Preprocessing and Model Ensemble on Twitter Sentiment Classification

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Steiner	Dominic
Wang	Yiming
Zhang	Yufei
Hong	Xin

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, July 31st 2023	
	Yiming Wang
	Yufei Zhang
	Xin Hong

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.