



Enhanced Sentiment Analysis of Social Media Text Using Transformer-Based Sarcasm Detection

Sudhir Kumar¹, Bharti Mittal², Sangeeta Rani³ * and Isha Sharma⁴

¹ Department of Computer Science and Engineering, World College of Technology and Management, Gurgaon, HR, India
Sudhirkr11032002@gmail.com

² Department of Computer Science and Engineering, World College of Technology and Management, Gurgaon, HR, India
mittal2.bharti@gmail.com

³ Department of Computer Science and Engineering, World College of Technology and Management, Gurgaon, HR, India (* Corresponding Author)
drsangeeta0909@gmail.com

⁴ Department of Computer Science and Engineering, World College of Technology and Management, Gurgaon, HR, India
ishu3072@gmail.com

Abstract. NLP has been one of the best frameworks used in text era especially in today's world of social media and influence. Sentiment analysis though old is still one of the most used tools; however, with increase of sarcasm in social media, traditional systems are failing to accurately understand and interpret. The research proposes a Transformer based model helping in understanding sarcasm which is crucial in sentiment analysis. To understand their clients and comments, many institutions use sentiment analysis. However, the increase in sarcasm has adversely affected sentiment analysis' efficiency and legitimacy. The proposed system will help elevate the validity and legitimacy of sentiment analysis by integrating sarcasm detection, adding trust to the results.

Keywords: Sentiment analysis, BERT, NLP, Sarcasm detection, Recommendation systems.

1 Introduction

There has been an explosion of data and comments on social media over the past few years, flooding the internet with public opinions and conversations. Enormous number of thoughts, reactions and experiences are posted each day. This has become a flood of data which can be put to use, providing valuable insights for many institutions which can be used in a number of ways which cannot be even imagined. This data helps business, government and researchers to grow and learn. Major technology though being ancient, NLP (natural language processing) [1] helps in processing this text data into valuable insights leading to practices of sentiment analysis, topic modeling et cetera. These practices help in reshaping this data which

helps in decision making. Sentiment analysis helps in identifying the sentiment behind a text and classifying the text into multiple classes depending upon the need. The prime example can be seen in YouTube studio which, in this context, a definite yes, a definite no, a neutral yes, and a definite no are the five major categories. Sentiment analysis not only in YouTube but also platforms like Amazon is being used, helping in recommendation and improvements.

Recent advancement in Deep Learning especially in Transformers, specially in Encoder Style Transformer with prime illustration of [2] BERT (Bidirectional Encoder Representation of Transformer), helps in understanding languages and relationships behind the words. Attention Mechanism (MHA: Multi head Attention) allows these models to understand relationships behind the words making it easier to understand complex sentences and its true meaning.

Sarcasm has highly affected NLP thus results from its applications are not that relevant and trusting despite metrics performance. In this research, we propose a sentiment analysis system that also takes sarcasm into account, reassuring results of sentiment analysis. The proposed system combines a BERT based sentiment analysis system with a sarcasm [3] detection model. The system first identifies sentiment of the given text and then checks whether it contains sarcasm or not. If sarcasm is detected then the result is modified accordingly, making results trusting and more accurate.

It consists of the following main pillars: BERT-based transformer sentiment analysis, an algorithm to detect sarcasm in social media, an integrated approach to detecting sarcasm combined with sentiment analysis for increased classification accuracy, analyzing benchmark datasets to assess the proposed model's performance.

IMDB movie reviews were used to train the sentiment classification model [4], which contains thousands of reviews labeled as positive or negative. For sarcasm detection, the TweetEval irony dataset was used. There are several types of social media posts in this dataset, including those categorized as sarcastic or non-sarcastic. The implementation of the models was performed using Python in a Google Colab environment. Libraries such as PyTorch and HuggingFace Transformers were used to fine-tune the BERT model for both tasks. During experimentation, the sentiment model [5] and sarcasm detection model were trained separately and then integrated into a combined pipeline. In the proposed system, the input text is first analyzed and modeled based on sentiment. After that, the sarcasm detection model checks whether the sentence contains sarcasm. If sarcasm is detected, the predicted sentiment polarity is adjusted so that the final output better reflects the actual meaning of the sentence.

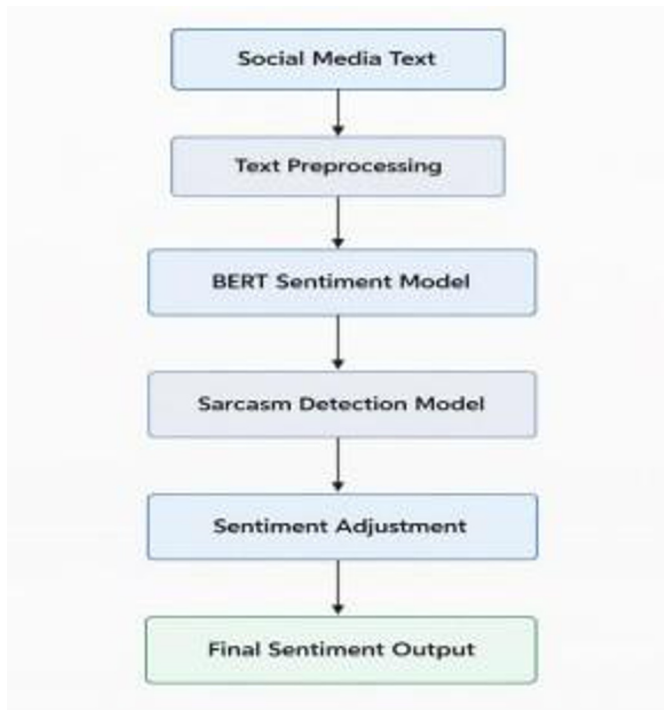


Fig. 1. Systematic Flow. Architecture of the proposed sarcasm-aware sentiment analysis system.

This study aims to enhance sentiment by enhancing [6] through the use of a translation. Sarcasm can be detected in message analysis models based on regression. By identifying sarcastic expressions before determining sentiment polarity, the system attempts to better capture the actual meaning behind social media text.

Table 1 presents a summary of previous research related to sentiment analysis and sarcasm detection.

Based on the literature review, it can be seen that transformer based models such as BERT have significantly improved the performance of sentiment analysis systems.

2 Literature Review

Sarcastic comments have been detected using sentiment analysis with both a machine learning and deep learning approach by various studies. Traditional classification algorithms work in classification of content but lack understanding of the context.

Recent advancement in deep learning, especially in transformers, improved the ability to understand content, thus improving efficiency in text tasks and adding the ability of capturing token relationships with each other, which earlier techniques significantly lacked.

Research contributions and studies in context to sentiment analysis, sarcasm detection and NLP are as follows:

Table 1. Summary of Notable NLP Research Contributions

Author	Year	Method	Dataset	Result
Vaswani et al.	2017	Transformer	WMT	Introduced attention-based N
Devlin et al.	2019	BERT	GLUE	Improved contextual langu
Ghosh & Veale	2016	Neural Network	Twitter	Detected sarcasm using con
Liu	2012	Sentiment Analysis	Reviews	Early opinion mining techni

BERT is significantly an important aspect of sentiment analysis, improving its [7] performance compared to traditional classification methods. However, lack of understanding sarcasm around social media is still an issue. Thus, integration of sentiment analysis with sarcasm detection is key for accuracy in real world legitimate applications.

3 Methodology

The entire system is broken down into multiple stages including text [8] preprocessing, classification using BERT, sarcasm utility and final sentiment result modification.

The approach is to improve accuracy of sentiment analysis by accounting for sarcastic expressions intended to alter the content.

Input data consists of tweets and user comments, Twitter being one of the most sarcastic platforms. Data is preprocessed using Re (an NLP library) which involves removing unwanted elements such as URLs, special characters et cetera, ensuring relevant information only.

In order to assess and predict the performance of the model, the cleaned text is tokenized using the BERT tokenizer.

After preprocessing, the [9] cleaned text is tokenized using the BERT tokens for prediction and training. Through BERT-generated contextual embeddings, the sentiment classification model predicts the sentiment polarity of an input text.

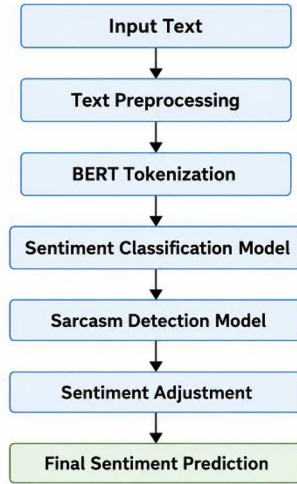


Fig. 2. Workflow of the proposed sarcasm-aware sentiment analysis system.

The probability of a sentiment class can be represented as:

$$P(y | x) = \text{Softmax}(W \cdot \text{BERT}(x) + b) \quad (1)$$

where x represents the input text, $\text{BERT}(x)$ represents the contextual embedding generated by the transformer model, and W and b are trainable parameters.

Sarcasm detection is performed using a classification function that determines whether a sentence is sarcastic or not:

$$S = f(x) \quad (2)$$

where S represents the sarcasm prediction and x represents the input text.

The final sentiment is adjusted based on sarcasm detection as follows:

$$\text{FinalSentiment} = 1 - \text{Sentiment}_{pred}, \text{ if } \text{sarcasm} = 1; \text{ Sentiment}_{pred}, \text{ otherwise} \quad (3)$$

The proposed system consists of two transformer-based models designed to [10] perform sentiment analysis and sarcasm detection independently before combining their outputs to generate the final sentiment prediction.

The sentiment classification model was trained using data from the IMDB movie reviews, while the sarcasm detection model was trained using the TweetEval irony dataset obtained from publicly available NLP benchmark repositories.

3.1 Dataset Description

Two datasets are used for the classification and sarcasm detection tasks. The IMDB dataset is used for the classification task that includes 50,000 labeled reviews, whereas for the sarcasm model the TweetEval irony dataset is used, consisting [11] of social media posts labeled as sarcastic or non-sarcastic.

Both the datasets are balanced in category. The dataset allows the model to learn contextual patterns commonly associated with sarcastic language in online communication. The training parameters used for fine-tuning the BERT models are summarized in Table 2.

Table 2. Training Parameters Used in the Model

Parameter	Value
Model Architecture	BERT-base-uncased
Framework	PyTorch
Library	HuggingFace Transformers
Batch Size	16
Learning Rate	2e-5
Optimizer	AdamW
Epochs	3
Max Sequence Length	128
Training Platform	Google Colab GPU

3.2 Sentiment Classification Model

The classification model is trained on the IMDB dataset. The dataset includes a [12] large number of reviews labeled either as positive or negative. The model used for the task is BERT base uncased.

By the end of training, the model understands the relationships and patterns leading to identifying whether the content is positive or negative.

3.3 Sarcasm Detection Model

The sarcasm model is [14] trained on the TweetEval irony dataset; the dataset includes social media comments labelled as sarcastic or non-sarcastic. Same as for classification, BERT Base Uncased is used for the task to maintain consistency and performance.

By the end of training, the model understands the relationships and patterns leading to [15] identifying whether the content is sarcastic or non-sarcastic.

3.4 Combined Pipeline

After training the sarcasm and classification models, both models are combined into unity, thus changing the flow of commands, and the [13] data is now parsed through the sarcasm model and then to the classification model.

If the sarcasm model labels the content as sarcastic, it now directly affects the produced sentiment from the classification model, making the predictions more realistic and accurate, boosting performance.

4 Results and Analysis

An experimental evaluation of the proposed system to analyze sentiment with sarcasm is presented in this section [13]. A Google Colab environment was used to train and evaluate the models using Python and deep learning libraries. In addition, accuracy metrics were used to quantify the performance of the sentiment analysis model, sarcasm detection model, and the combined sarcasm aware system.

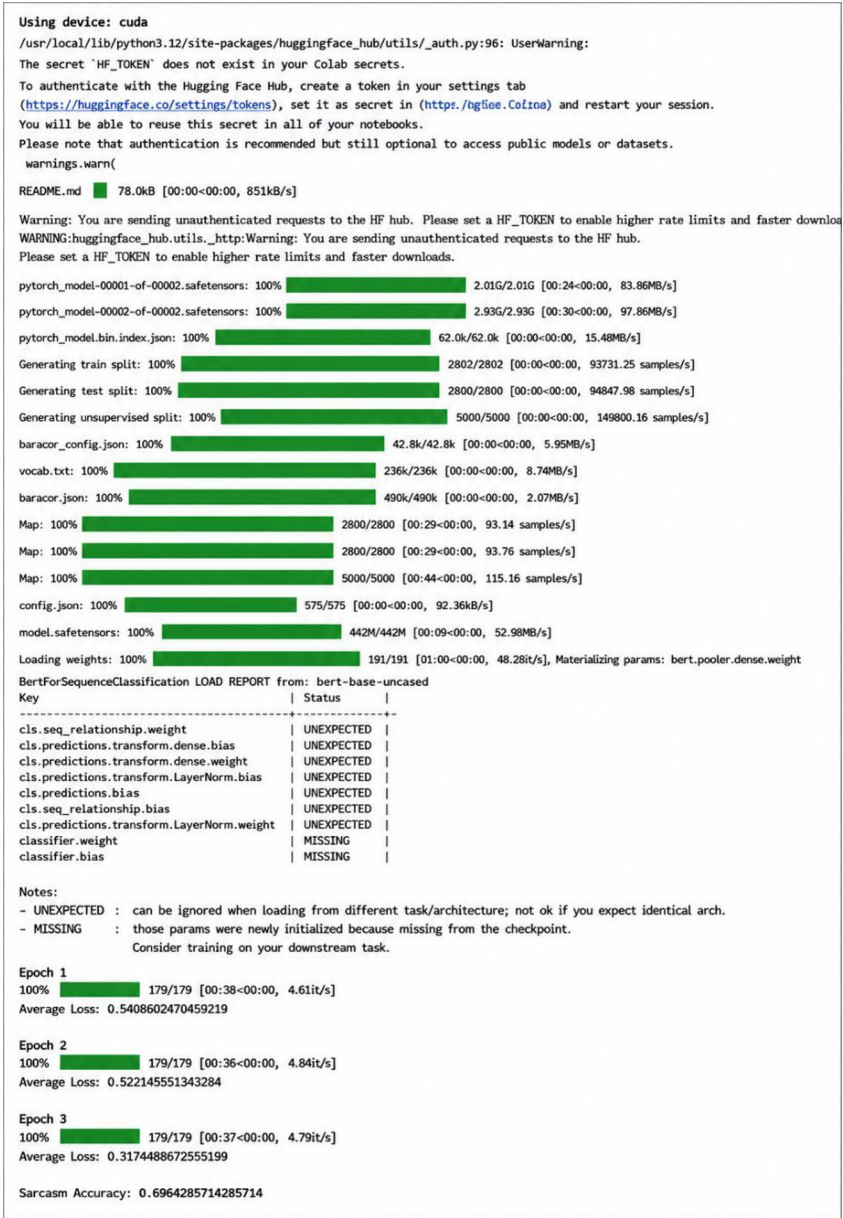


Fig. 3. Training Process and Accuracy.

```

Using device: cuda
README.md: 23.0% [00:00<00:00, 2.24MB/s]
ironytrain-00000-of-00001.parquet: 100% [183/183 [00:01<00:00, 148MB/s]
ironytest-00000-of-00001.parquet: 100% [54.0/54.0k [00:00<00:00, 123kB/s]
ironyvalidation-00000-of-00001.parquet: 100% [61.1k/61.1k [00:00<00:00, 133kB/s]
Generating train split: 100% [2862/2862 [00:00<00:00, 96842.39 examples/s]
Generating test split: 100% [784/784 [00:00<00:00, 32572.28 examples/s]
Generating validation split: 100% [955/955 [00:00<00:00, 49476.41 examples/s]
Map: 100% [2862/2862 [00:00<00:00, 9142.25 examples/s]
Map: 100% [784/784 [00:00<00:00, 7452.32 examples/s]
Map: 100% [955/955 [00:00<00:00, 7382.78 examples/s]
Loading weights: 100% [191/191 [00:00<00:00, 648.48it/s, Materializing param=bert.pooler.dense.weight]

BertForSequenceClassification LOAD REPORT from: bert-base-uncased
Key | Status |
-----|-----|
cls.seq_relationship.weight | UNEXPECTED |
cls.predictions.transform.dense.bias | UNEXPECTED |
cls.predictions.transform.dense.weight | UNEXPECTED |
cls.predictions.transform.LayerNorm.bias | UNEXPECTED |
cls.predictions.bias | UNEXPECTED |
cls.seq_relationship.bias | UNEXPECTED |
cls.predictions.transform.LayerNorm.weight | UNEXPECTED |
classifier.weight | MISSING |
classifier.bias | MISSING |

Notes:
- UNEXPECTED : can be ignored when loading from different task/architecture; not ok if you expect identical arch.
- MISSING : these params were newly initialized because missing from the checkpoint.
            Consider training on your downstream task.

Epoch 1
100% [179/179 [00:38<00:00, 4.61it/s]
Average Loss: 0.5408602470459219

Epoch 2
100% [179/179 [00:36<00:00, 4.84it/s]
Average Loss: 0.522145551343284

Epoch 3
100% [179/179 [00:37<00:00, 4.79it/s]
Average Loss: 0.3174488672555159

Sarcasm Accuracy: 0.6964285714285714

```

Fig. 4. Accuracy of the sarcasm detection model.

Table 3. Performance Comparison of Models

Model	Accuracy (%)
Sentiment Analysis Model (BERT)	85.1
Sarcasm Detection Model (BERT)	69.6
Combined Sentiment Pipeline	49.1

```

Using device: cuda
Loading weights: 100% ██████████ 201/201 [00:00<00:00, 622.84it/s, Materializing param=classifier.weight]
Loading weights: 100% ██████████ 201/201 [00:00<00:00, 767.37it/s, Materializing param=classifier.weight]

Text: I love this phone
Sarcasm: 1
Final Sentiment: 0

Text: This movie is terrible
Sarcasm: 0
Final Sentiment: 0

Text: Great. Another Monday.
Sarcasm: 1
Final Sentiment: 0

Text: Oh fantastic, my exam got postponed again
Sarcasm: 1
Final Sentiment: 0

```

Fig. 5. Proposed System Prediction.

The result indicates a sharp decline in accuracy of the model when integrated with [16] sarcasm awareness; the sarcasm polarity is one of the key factors behind the sharp decline in accuracy though when tested independently both the models' performance was good enough.

5 Discussion

Both of the tasks, whether classification or sarcasm detection, achieved good accuracy independently, showcasing the performance and agility of [17] Transformer based classification, capturing relationships and underlying patterns in the context, far surpassing traditional machine learning methods.

The combined system illustrated [18] a sharp decline in the accuracy; the metrics show low accuracy on the tasks but, in the end, the predictions made were accurate at least for sarcastic content.

Some key reasons behind the sharp decline could be improper balance between models' decisions, [19] sarcasm polarity indifference and some other integrating factors.

Despite the limitations, adding [20] sarcasm awareness adds another layer of complexity to the task, though necessary for real world application and deployment.

6 Conclusion

The experiment allows us to understand the severity of sarcasm for applications like sentiment analysis and content modeling; with the increase in sarcastic content over the internet, it has become extremely important to account for it in measures.

Sentiment analysis is an important aspect of deep learning and when infused with sarcasm detection makes its results more reliable. Though integration gaps and methods are still a major concern, as the method used in the experimentation indicates a sharp decline in metrics which is confusing.

The polarity of sarcasm infused with the results of the sentiment model could be more finely integrated, and with better integration the metrics should rise. Two major factors for such decline in metric are due to improper integration and polarity of sarcasm.

7 Future Directive

Future work will be focused on improving generalization and reliability for integration of modern AI methods to create a balance between the integration of sarcasm and classification task to improve reliability of sentiment analysis.

References

1. A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems*, 2017.
2. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL*, 2019.
3. A. Ghosh and T. Veale, "Fracking Sarcasm using Neural Network," *Proceedings of ACL*, 2016.
4. B. Liu, "Sentiment Analysis and Opinion Mining," Morgan & Claypool Publishers, 2012.
5. Y. Zhang, Q. Jin, and S. Zhao, "Deep Learning for Sentiment Analysis," *IEEE Transactions on Knowledge and Data Engineering*, 2022.
6. E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, "A Practical Guide to Sentiment Analysis," Springer, 2017.
7. T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," *ICLR*, 2013.
8. A. Radford et al., "Language Models are Unsupervised Multitask Learners," *OpenAI Technical Report*, 2019.
9. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
10. Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proceedings of EMNLP*, 2014.
11. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the ACL*, 2017.
12. J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," *EMNLP*, 2014.

13. S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4: Sentiment Analysis in Twitter," SemEval, 2017.
14. M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-Based Methods for Sentiment Analysis," Computational Linguistics, 2011.
15. A. Joulin et al., "Bag of Tricks for Efficient Text Classification," Proceedings of EACL, 2017.
16. S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," arXiv preprint arXiv:1706.05098, 2017.
17. M. Peters et al., "Deep Contextualized Word Representations," NAACL, 2018.
18. T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," Proceedings of EMNLP, 2020.
19. J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," ACL, 2018.
20. I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

