

# Sentiment Analysis Using State of the Art Machine Learning Techniques

Salih Balci[1], Gozde Merve Demirci[2], Hilmi Demirhan[3], and Salih Sarp[4(✉)]

[1] Gebze Technical University, Kocaeli, Turkey
s.balci2020@gtu.edu.tr
[2] The Graduate Center CUNY, New York, USA
gdemirci1@gradcenter.cuny.edu
[3] University of North Carolina, Chapel Hill, USA
hd1526@uncw.edu
[4] Virginia Commonwealth University, Richmond, USA
sarps@vcu.edu

**Abstract.** Sentiment analysis is one of the essential and challenging tasks in the Artificial Intelligence field due to the complexity of the languages. Models that use rule-based and machine learning-based techniques have become popular. However, existing models have been under-performing in classifying irony, sarcasm, and subjectivity in the text. In this paper, we aim to deploy and evaluate the performances of the State-of-the-Art machine learning sentiment analysis techniques on a public IMDB dataset. The dataset includes many samples of irony and sarcasm. Long-short term memory (LSTM), bag of tricks (BoT), convolutional neural networks (CNN), and transformer-based models are developed and evaluated. In addition, we have examined the effect of hyper-parameters on the accuracy of the models.

**Keywords:** Sentiment analysis · Bag of tricks · Transformer · BERT · CNN

## 1 Introduction

Over a decade, so many users have generated lots of content on the internet, mostly on social platforms. Millions of individuals use any social platform to express their opinions, and these contents create a considerable amount of raw data. Such a massive volume of raw data brings lots of exciting tasks. These tasks are under Natural Language Processing (NLP) applications. NLP is a branch of Artificial Intelligence (AI) focusing on text-related problems, and one of its goals is to understand the human language. In NLP, there are various language-related fields to focus on, such as machine translation, chatbots, summarizations, question answering, sentiment analysis [1, 2].

Sentiment Analysis (SA) is closely linked to NLP. Sentiment analysis is a scale result that shows the sentiment and the opinions coming from a raw text. It is essential and helpful application to analyze an individual's thoughts. The sentiment analysis result may help various fields from industrial purposes such as advertising and sales to academic

© The Author(s) 2022
C. Biele et al. (Eds.): MIDI 2021, LNNS 440, pp. 34–42, 2022.
https://doi.org/10.1007/978-3-031-11432-8_3

purposes. Even though sentiment analysis has been a focus of authors for a while, the challenges in this field, such as having sarcasm and irony in a text, make this task still unfinished. Therefore, there is still colossal attention on sentiment analysis, and new approaches have arisen [3].

Recently, many novel approaches to AI systems have been developed using Machine Learning (ML). Also, with the help of Deep Learning (DL) techniques, a subfield of ML, the algorithms such as Generative Adversarial Networks and transformers improved the performance of AI tasks significantly [4]. Many studies have focused on sentiment analysis in NLP fields [5–7]. Today, comprehensive survey studies and novelty approaches to sentiment analysis are still being carried out. In [5], the authors created an extensive research survey on sentiment analysis. In the paper, levels of sentiment analysis, challenges and trends in this field, and the genetic process are mentioned detail. Here, sarcasm detection was shown as one of the challenges, and related studies to solve this challenge are examined. Instead of traditional machine learning approaches, other techniques such as DL and reinforcement learning resulted in more robust solutions to challenges.

Authors in [6] proposed a hybrid model by combining the DL approach and sentiment analysis model to predict the stock prices. Sentiment analysis in the stock market is critical to estimating future price changes. In this article, the authors created a hybrid model using a Convolutional Neural Network (CNN) to create a sentiment analysis classifier on investors' comments and Long Short-Term Memory (LSTM) Neural Network to analyze the stock. Implementation of this hybrid model on the real-life data on the Shanghai Stock Exchange (SSE) showed that the hybrid approach outperformed.

In the paper [7], the study conducts a novel approach to ML-based classifiers. From Twitter, related tweets have been retrieved from eight countries, and people's behavior on the infectious disease was aimed to analyze. In the proposed model, Naïve Bayes Support Vector Machines (NBSVM), CNN, Bidirectional Gated Recurrent Network (BiGRU), fastText, and DistilBERT [8] were used as base classifiers, and the fusion of these approaches was represented as "Meta Classifier". The proposed model gave better results than four DL and one machine learning approach.

This paper gives the comparison works on sentiment analysis using state-of-art ML approaches: LSTM, Bag of Words (BoT), CNN, and transformer. The aim is to compare the performances of deep learning approaches in terms of accuracy and time complexity. Moreover, the impact of hyperparameters on the model's accuracy was analyzed.

This paper is organized as follows. In the second section, background information about sentiment analysis, particularly in NLP, is discussed. In the third section, the approach to the methodology and the state-of-art approach are explained. The fourth section explains the results, and the paper is concluded in the last section.

## 2 Background Information

### 2.1 Preprocessing the Text Data

The abundance of text data provides many opportunities for training the NLP models. However, the unstructured nature of the text data requires preprocessing. Lowercasing, spelling corrections, punctuation, and stop word removal are some of these preprocessing steps. These operations could be easily implemented in Python language using NumPy,

pandas, textblob, or nltk libraries. Then, tokenization, stemming, and lemmatization processes are realized to convert raw text data to smaller units with removing redundancy. The tokenization process splits the stream of text into words [9]. Extracting the root of a word is done using stemming techniques. Similar to stemming, the lemmatizing process extracts the base form of a word. These preprocessing steps could be implemented using textblob or nltk libraries as well. The data used in this study are taken from the public IMDB dataset. It has binary labeled 50000 reviews. To train the SA models, 17500 reviews are chosen for training, 7500 for validation, and 25000 for testing purposes.

### 2.2 Feature Engineering

After preprocessing, feature extraction steps are implemented to transform words or characters to computer understandable format. This step includes vectorization. Vectorization of words provides corresponding vectors for further processing. Relating the representations of the words with similar meanings is achieved using word embedding. Each word is represented with a vector using feature engineering methods such as N-grams, count vectorizing, and term frequency-inverse document frequency (TF-IDF). Word embedding methods are developed to capture the semantic relation between the words [10]. Word2vec and fastText frameworks are developed to train word embeddings. Skip-Gram and Continuous Bag of Words (CBOW) are commonly used models [11]. This study benefits from the pre-trained GloVe word embeddings to increase the performance of the sentiment analysis models.

## 3 Methodology

This section develops four major sentiment analysis models using ML techniques, i.e., LSTM, BoT, CNN, and transformer.

### 3.1 LSTM-Based Sentiment Analysis

Neural networks utilize backpropagation algorithms to update the weights using chain rule in calculus. For large deep neural networks, backpropagation could cause troubles such as vanishing or exploding gradients. Long-term memory architecture is an improved version of the Recurrent Neural Network (RNN) to overcome the vanishing gradient problem with an extra recurrent state called a memory cell. LSTM achieves long-range data series learning, making it a suitable technique for a sentiment analysis task.

Forward and backward RNNs are combined to form a single tensor to increase the performance of the LSTM-based model. In addition to bidirectionality, multiple LSTM layers could be stacked on top of each other to increase the performance further. We have used two layers of LSTM and 0.5 dropout on hidden states for regularization to decrease the probability of overfitting.

### 3.2 Bag of Tricks-Based Sentiment Analysis

Among DL techniques, linear classifiers could achieve similar performances with a more straightforward design [12]. However, one of the disadvantages of linear classifiers is their inability to share the parameters among features and classes [12]. The Bag of Tricks (BoT) architecture uses linear models with a rank constraint and a fast loss approximation as a base. Means of words' vector representations are fed to linear classifiers to get a probability distribution of a class. In this study, bag of n-grams, i.e., bigrams, are utilized instead of word order for higher performance where the n-gram technique stores n-adjacent words together.

This architecture also does not use pre-trained word embeddings, which could ease its usage in other languages that do not yet have efficient pre-trained word embeddings. This model has fewer parameters than the other models, and the results are comparable in less time.

### 3.3 CNN (Convolutional Sentiment Analysis)-Based Sentiment Analysis

Convolutional Neural Network (CNN) is a DL approach used primarily for raw data. CNN has a wide scope of application fields from image recognition to NLP with its architecture [13, 18, 19]. CNN has a multi-layered feed-forward Neural Network architecture. It aims to reduce the data into a shape so that it does not lose any features while processing. This way, CNN makes sure that the prediction accuracy and quality will be higher. There are convolutional and pooling layers in its architecture to reshape the data while training. Traditionally, CNNs have one or more convolutional layers followed by one or more linear layers.

In the convolutional layer, the data is processed and reshaped by kxk filters, usually k = 3. Each filter has a form in the architecture, and this filter gives the weight for the data points. The intuitive idea behind learning the weights is that the convolutional layers act as feature extractors, extracting parts of the most critical data. This way, the dominant features are extracted.

CNN has been mainly used for image fields for a long time; image recognition, image detection, analysis, etc. However, it is started to be used in NLP approaches, and it gives significant results. In this study, convolutional layers will be used as "k" consecutive words in a piece of text. The kxk filter mentioned in the convolutional layer would represent a patch of an image in image related field. However, here 1xk filter will be used to focus on k consecutive words as in bi-grams (a 1x2 filter), tri-grams (a 1x3 filter), and/or n-grams (a 1xn filter) inside the text.

### 3.4 Transformer Based Sentiment Analysis

The transformer is a state of art network architecture proposed in 2017 [14]. In this state-of-art approach, the results showed that with the use of a transformer, NLP tasks outperformed other techniques. After transformer architecture, various models focusing on NLP fields such as ROBERT [15], BERT [16], ELECTRA [17] were proposed. Specifically, BERT (Bidirectional Encoder Representations from Transformers) model is one of the most robust state-of-art approaches on NLP fields. BERT was introduced

in 2019 by Google AI Language, and since then, it has started to be used very quickly in academics and industry. BERT is a pre-trained model which is very easy to fine-tune model into our dataset. It has a wide range of language options [16].

BERT architecture is a multi-layer bidirectional transformer encoder. BERT input representation has three embedding layers: position embedding, segment embedding, and token embedding. In the pre-training part, BERT applied two unsupervised tasks, Masked LM (MLM) and Next Sentence Prediction (NSP), instead of traditional sequence modeling. BERT has pre-trained with more than 3,000 M words.

In the study, we used the transformers library to obtain a pre-trained BERT model employed as our embedding layers. We only trained the rest of the model with the pre-trained architecture itself, which learns from the transformer's representations. The transformer provides a pooled output and the embedding for the whole sequence. Due to the purpose of this study (sentiment outputs), the model has not utilized the pooled output from the architecture.

The input sequence of the data was tokenized and trimmed to the maximum sequence size. The tokenized input was converted to a tensor and prepared for fine-tuning. After fine-tuning the model, it was then used to evaluate the sentiment of various sequences.

## 4 Results and Discussion

This section compares performances of state-of-the-art (SOTA) models in terms of accuracy, time, and loss.

### 4.1 Time Analysis

The training time comparisons of SOTA models are indicated in Table 1. The results indicated that most DL models provide reasonable training time except the transformer-based model. Models that use LSTM, BoT, and CNN performed an epoch per minute, whereas the BoT-based model achieves 13 s per epoch in contrast to 28 min in the case of the transformer model. In the testing phase, results are aligned with the training phase. Even though only time analysis does not give a concrete interpretation of a model, we see a considerable time efficiency difference between BERT and other models.

**Table 1.** Training and testing time comparison of SOTA models

| Epoch/Test | LSTM | BoT | CNN | Transformer |
|---|---|---|---|---|
| 1 | 1 m 41 s | 0 m 14 s | 0 m 30 s | 28 m 4 s |
| 2 | 1 m 40 s | 0 m 13 s | 0 m 30 s | 28 m 7 s |
| 3 | 1 m 41 s | 0 m 13 s | 0 m 30 s | 28 m 6 s |
| 4 | 1 m 40 s | 0 m 13 s | 0 m 30 s | 28 m 7 s |
| 5 | 1 m 40 s | 0 m 13 s | 0 m 30 s | 28 m 7 s |

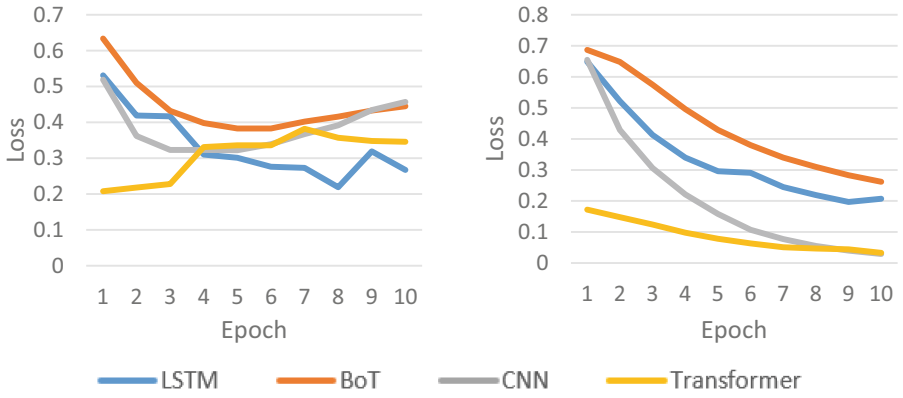
(*continued*)

**Table 1.** (*continued*)

| Epoch/Test | LSTM | BoT | CNN | Transformer |
|---|---|---|---|---|
| 6 | 1 m 41 s | 0 m 13 s | 0 m 30 s | 28 m 6 s |
| 7 | 1 m 41 s | 0 m 13 s | 0 m 30 s | 28 m 4 s |
| 8 | 1 m 40 s | 0 m 13 s | 0 m 30 s | 28 m 7 s |
| 9 | 1 m 40 s | 0 m 13 s | 0 m 30 s | 27 m 56 s |
| 10 | 1 m 40 s | 0 m 13 s | 0 m 30 s | 27 m 58 s |
| Testing | 15 ms | 9 ms | 10 ms | 35 ms |

### 4.2 Validation and Test Losses

Validation loss is another critical metric to evaluate how a model fits new data. Validation loss is also a good indicator of overfitting. The models' validation, training, and test losses are shown in Fig. 1 and Table 2.




a) Validation Loss b) Training Loss

**Fig. 1.** Validation and training losses of the models.

The loss graph of the transformer-based model indicates that it could converge faster than other models with fewer training epochs. This will be a result of pre-training of the transformer model.

**Table 2.** Test losses of the models.

| Models | LSTM | BoT | CNN | Transformer |
|---|---|---|---|---|
| Loss | 0.323 | 0.391 | 0.344 | 0.209 |

### 4.3 Validation Accuracy

Validation accuracy in combination with validation loss could be used to determine the model's generalization ability. The validation and testing accuracies of the models are given in Table 3. Validation accuracy reveals that five epochs of training are enough to get good results which are also in line with the validation loss. Testing accuracy is aligned with the validation accuracy where the transformer-based model achieves the best performance.

**Table 3.** Validation and testing accuracies of the models.

| Epoch/Test | LSTM | BoT | CNN | Transformer |
|---|---|---|---|---|
| 1 | 73.28% | 72.08% | 77.03% | 91.93% |
| 2 | 82.19% | 76.29% | 84.56% | 91.76% |
| 3 | 79.64% | 80.50% | 86.06% | 92.02% |
| 4 | 87.71% | 83.55% | 86.78% | 90.74% |
| 5 | 87.81% | 85.47% | 86.99% | 91.31% |
| 6 | 89.27% | 85.47% | 87.23% | 91.31% |
| 7 | 89.65% | 87.09% | 87.16% | 90.89% |
| 8 | 91.52% | 87.68% | 87.30% | 91.19% |
| 9 | 88.06% | 88.07% | 86.96% | 92.15% |
| 10 | 89.69% | 88.46% | 87.40% | 91.85% |
| Testing | 86.96% | 85.18% | 85.04% | 91.58% |

Observations derived from the performance comparisons are outlined below.

Observation 1: BoT-based model is faster than other DL models.

Observation 2: Transformer-based model takes a long time to train and predict.

Observation 3: Optimum epoch number could be determined using accuracy and loss of training and validation phases. Five epochs of training provide optimum training.

Observation 4: Transformer-based model converges faster than other models.

## 5 Conclusion

Sentiment analysis has been studied to harness the reviews, comments, and other written documents. The potential of sentiment analysis provided many benefits to various industries such as entertainment and e-commerce. This paper presents sentiment analysis models that utilize four ML techniques, i.e., LSTM, BoT, CNN, and transformer. Their performances in terms of time, loss, and accuracy are examined and compared. The BoT-based sentiment analysis model is faster than other ML models, whereas the transformer-based model performs poorly in terms of time. Furthermore, this study also demonstrates the accuracies of these models. The transformer-based sentiment analysis model achieved higher accuracy than other ML models.

This study indicates that ML techniques could be utilized successfully for sentiment analysis tasks. It is expected that this study will be helpful for both developers and researchers while deploying ML-based sentiment analysis algorithms into their projects.

## References


1. Balki, F., Demirhan, H, Sarp, S.: Neural machine translation for Turkish to English using deep learning. In: Biele, C., Kacprzyk, J., Owsiński, J.W., Romanowski, A., Sikorski, M. (eds.) Conference on Multimedia, Interaction, Design and Innovation. AISC, vol. 1376, pp. 3–9. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-74728-2_1
2. Demirci, G.M., Keskin, Ş.R., Doğan, G.: Sentiment analysis in Turkish with deep learning. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 2215–2221. IEEE (2019)
3. Farzindar, A., Inkpen, D.: natural language processing for social media. Syn. Lect. Hum. Lang. Technol. **8** 1–166 (2015)
4. Sarp, S., Kuzlu, M., Wilson, E., Guler, O.: WG2AN: synthetic wound image generation using generative adversarial network. J. Eng. **2021**(5), 286–294 (2021)
5. Birjali, M., Kasri, M., Beni-Hssne, A.: A comprehensive survey on sentiment analysis: approaches, challenges and trends. Knowl. Based Syst. **226**, 107134 (2020)
6. Jing, N., Wu, Z., Wang, H.: A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. Expert Syst. Appl. **178**, 115019 (2021)
7. Basiri M.E., Nemat, S., Abdar, M., Asadi, S., Acharrya, U.R.: A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets. Knowl. Based Syst. **228** (2020)
8. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
9. Vijayarani, S., Janani, R.: Text mining: open source tokenization tools-an analysis. Adv. Comput. Intell. Int. J. **3**(1), 37–47 (2016)
10. Kulkarni, A., Shivananda, A.: Natural language processing recipes. Apress (2019)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
12. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
13. Sarp, S., Kuzlu, M., Pipattanasomporn, M., Guler, O.: Simultaneous wound border segmentation and tissue classification using a conditional generative adversarial network. J. Eng. **2021**(3), 125–134 (2021)
14. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
15. Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y., Dehak, N.: December. Hierarchical transformers for long document classification. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 838–844. IEEE (2019)
16. Devlin J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL (2019)
17. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
18. Sarp, S., Kuzlu, M., Wilson, E., Cali, U., Guler, O.: The enlightening role of explainable artificial intelligence in chronic wound classification. Electronics **10**(12), 1406 (2021)
19. Yin, W., Kann, K., Yu, M., Schütze, H.: Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923 (2017)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits 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.

