SENTIMENT ANALYSIS ON SOCIAL MEDIA CONTENT

Boris Borovčanin

Department of Information Technologies, Faculty of Engineering, Natural and Medical Sciences, International Burch University, Sarajevo, Bosnia and Herzegovina, boris.borovcanin@stu.ibu.edu.ba, ORCID ID: 0009-0002-7993-0544

Original scientific paper

https://doi.org/10.7251/JIT2601040B UDC: 004.738.5:[658.8:659.1

Abstract: Following research evaluated conventional machine learning and deep learning algorithms used for the purpose of binary text classification, in accordance with previous research demonstrating advantages in supervised learning models such as Naive Bayes, Logistic Regression, and LSTM networks. Models that were subject of implementation are: Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, and LSTM. Responses from nonprofit organizations have been cleaned, tokenized, and preprocessed implementing either TF-IDF vectorization or sequence trimming determined by the model that was chosen. The majority of the models were performed using 50,000 samples because of computational capacity limitations, whereas the LSTM was executed only with 5,000 samples. LinearSVC is implemented for the purpose of accelerating training of the SVM model, as well as Random Forest parameters optimization for algorithmic efficiency. On the other hand the LSTM model provided an embedding component and a single LSTM unit for maintaining the sequence information. The performance of the models was evaluated according to the accuracy, precision, recall, and F1 score metrics. The findings are indicating that fundamental models perform effectively and consistently, however the LSTM model demands more computational capacity to provide context for classification.

Keywords: sentiment analysis, twitter data, machine learning, performance metrics

INTRODUCTION

Sentiment analysis or opinion mining is a widely studied application of Natural Language Processing (NLP) that involves identifying and classifying opinions contained in text data-specifically if the sentiment of a statement is positive, negative, or neutral. With user-generated content ballooning exponentially on Twitter, Facebook, Reddit, and Instagram, sentiment analysis has been rendered crucial in understanding public opinion, monitoring brand reputation, detecting misinformation, and even predicting political leanings.

In the age of the internet, people tend to share opinions online. Those unstructured bits of data have plenty of information but are hard to analyze at large manually. The overall interest in sentiment analysis is due to its applications across fields—businesses use it to improve user experience, governments monitor it for indicators of unrest, and researchers use it for studies of mental illness or consumer behavior.

The problem solved by this work is the analysis of social media sentiment data through the automatic classification of social media postings (such as tweets) to identify whether they have a positive or negative sentiment. In this research, posting classification used five different machine learning models to classify postings into binary (two classes) or multiclass (three classes) classifications. For this purpose, I have used the Sentiment140: Twitter Sentiment Analysis Dataset **[1]**, which contains 1.6 million annotated tweets labeled for sentiment classification.

The Sentiment140 dataset contains tweets assigned polarity sentiment classifications. An example of this is the positive tweet "Lyx is cool" that was created by user robotickilldozr on May 16, 2009, and is given a sentiment score of 4 (positive). The purpose of this example is to show how the dataset encodes the public's expression of feelings toward something into values that can be used for analyzing sentiment.

RELATED WORKS ON SENTIMENT ANALYSIS

Since there is a lot of user-generated content for expressing the general attitude of society at large it is crucial to highlight that social media sentiment anal-

ysis and statistics, especially data related to Twitter posts, have been a primary focus of interest in natural language processing (NLP). It is important to consider different strategies that have been utilized with a wide range of models including rule-based and conventional machine learning models on the one side, as well as transformer and deep learning models on the other. Although preliminary studies were focused on techniques for remote monitoring in large-scale data processing, research presented in **[1]**, it is important to emphasize that recent initiatives have included additional domain features and neural architectures to enhance sentiment classification efficiency. Even though significant advances have been made, concerns including flexibility of the domain, as well as the informal text, and absence of suitable labeled data in languages with limited resources remain in the focus of the corresponding studies.

Observing the approach established in **[1]** the foundation utilizing Sentiment140 data by leveraging emoticons as distant supervision labels (":\)" and ":\(") allowing scalable gathering of sentiment-labeled tweets. Corresponding study utilizes the same dataset as utilized in our project. Their methodology encouraged most modern social media sentiment processing pipelines.

Research presented in **[2]** talks about deep learning models (CNN, RNN, LSTM) used in sentiment analysis and highlights their advantages over traditional ML methods. This survey is relevant to the topic of my research since it provides theoretical grounding for your use of LSTM and justifies your choice of deep learning for modeling sentiment on unstructured text.

The main idea of work proposed in **[3]** paper surveys sentiment analysis techniques over Twitter data and suggests challenges such as short text, sarcasm, and colloquialisms. Paper is relevant to this topic since it provides assistance to establish the unique nature of Twitter sentiment analysis and helps the need to preprocess noisy tweets before classification.

When it comes to the idea introduced in **[4]** article provides a clear explanation of sentiment analysis algorithms (lexicon-based, ML-based) and areas of application like politics, business, and healthcare. Considering that article helps in defining a project's social impact and positions your model in the broader scope of application.

Although CNN-focused, the work discussed in **[5]** introduced new methods for using deep learning for text classification that set the stage for many subsequent sentiment models. Demonstrates how deep structures like CNNs (and later, LSTMs) can achieve better results than classic methods - comparative justification, relating it to the topic of my research.

Dataset preparation

In the manner of this study in the foreground is a comparison of five supervised learning models for binary text classification, including Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, and Long Short-Term Memory (LSTM) networks. Following that the pipeline began with a preprocessing stage which included tokenization, contaminating the text, and encoding the labels. Considering the traditional machine learning model, the feature extraction process was performed using TF-IDF vectorization.

In light of computer limitations and to accelerate selected models, only 50,000 samples were used to train all the models, except the LSTM model which was limited to 5,000 samples. LSTM stands out from other models because it could not handle textual data as a matrix of fragmented features. Keras's Tokenizer and pad_sequences were utilized in order to convert the text into sequences of augmented integers to be used by LSTM since it takes sequential inputs. Taking it into consideration all models were trained and tested according to different performance measures including: accuracy, precision, recall, and F1 score.

Data Preprocessing

Dataset has been broken down into training and testing sets for the purpose of performance evaluation on data that is not part of the sample. Each component of textual data encountered basic the preparation process, including:

- Stopword removal and punctuation
- Lowercase all tokens
- Stemming or lemmatization

The dataset was already in binary classification format, consequently there was no need for one-hot encoding or manual label encoding.

Text Vectorization

Three vectorization approaches have been addressed:

- TF-IDF (Term Frequency–Inverse Document Frequency): The majority of models supported this approach, which involved transforming text to a minimal matrix of normalized word frequencies.
- CountVectorizer: Converts a collection of text documents into a token counts matrix.
- HashingVectorizer: This method represents a more efficient, as well as memory-friendly method of vectorizing text that implements the hashing trick.

Considering how it impacts the memory and performance, TF-IDF has been chosen with `max_features=1000`, as the default for all the classical models.

RESEARCH DESIGN

Exploratory Data Analysis (EDA)

In advance of doing the analysis, the initial step was Exploratory Data Analysis (EDA), for the purpose of better dataset comprehension. This has been accomplished by implementing text lengths analysis, while discovering the most frequent words, and visualizing the class distributions. Additionally, the preprocessing stage included a wide range of tasks, such as identifying missing values, stopwords, as well as text normalization that has been applied in the same time frame. The most commonly used phrases in each class are presented as well using semantic clouds and bar charts, which enhanced the feature selection and allowed observations, which could be useful in understanding the data itself. Furthermore, EDA has provided assistance for easier decision making when it comes to tokenization and vectorizer implementation used afterwards in the stage related to development of the model.

Model Development

Five different algorithms were trained and implemented for binary classification of text. Each algorithm was chosen based on its applicability to the task, scalability, and ability to work with sparse or sequential data.

Logistic Regression was used as a baseline, linear classifier algorithm due to its speed, simplicity, and interpretability (Fig. 1). It had been trained with TF-IDF vectorizer features in order to preserve importance of the term while still being computationally efficient. The model would choose features that had a sufficient amount of importance while still disregarding terms of lower importance. Training the Logistic Regression model directly to a limited, high-dimensional matrix could result in relatively inefficient convergence. Furthermore, the maximum number of iterations completed was increased to ensure convergence while training.

```
# Logistic Regression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train_vec, y_train)
metrics['Logistic Regression'] = evaluate_
model("Logistic Regression", y_test,
lr.predict(X_test_vec))
```

Figure 1. *Logistic regression*

Naive Bayes algorithm (MultinomialNB specifically) was chosen for its stochastic structure, while resulting in outstanding performance with previous research in binary classifications of text (Fig. 2). Naive Bayes assumes independence among features, which is a potential simplification of reality, but effectively works in classification tasks where the features can be substantially represented in their states, either with word frequencies or terms method of TF-IDF. Since Naive Bayes is a lightweight architecture and performs very well with sparse representations of data, it was expected to be one of the most efficient models in the experiment.

```
# Naive Bayes
nb = MultinomialNB()
nb.fit(X_train_vec, y_train)
metrics['Naive Bayes'] = evaluate_
model("Naive Bayes", y_test, nb.predict(X_
test_vec))
```

Figure 2. *Naïve Bayes*

Support Vector Machine (SVM) was implemented using the SVC() classifier at first, but because it had slow training times and convergence issues with large, high-dimensional datasets, it was topologically replaced with LinearSVC. The linear version, while different theoretically, is more suitable for sparse and large-scale text classification tasks by providing more

scalability and implementation alternatives without significantly compromising robustness, such as accuracy. (Fig. 3)

```
# SVM
svm = LinearSVC(max_iter=1000)
svm.fit(X_train_vec, y_train)
metrics['SVM'] = evaluate_model("Linear SVM", y_test, svm.predict(X_test_vec))
```

Figure 3. *SVM*

Random Forest was also implemented because it is an ensemble learning method, therefore combining multiple decision trees to improve performance and robustness. There were considerable challenges with resources and an extensive training period since the problem was how it was used in the implementation. Following that, the model was tuned (n_estimators=10, max_depth=5, n_jobs=-1) to limit the tree number, tree depth, and enable parallel computations just to train a model efficiently while still capturing important feature interactions. (Fig. 4)

```
# Random Forest
rf = RandomForestClassifier(n_estimators=10, max_depth=10, random_state=42)
rf.fit(X_train_vec, y_train)
metrics['Random Forest'] = evaluate_model("Random Forest", y_test, rf.predict(X_test_vec))
```

Figure 4 *Random forest*

Long Short Term Memory (LSTM) neural network was used to explore deep learning-based sequential modeling. The model was built using TensorFlow/Keras. The text input was tokenized and represented as augmented integer sequences, which were converted to be input to the embedding layer and LSTM. In general, the architecture of the LSTM is done in a structure of first an embedding layer that takes tokens and converts them into dense vectors, followed by a single LSTM layer that captures temporal dependencies in sequences of words. Additionally, two layers were built including the regularization dropout layer as well as the final dense layer for binary classification with a sigmoid activation function. Because of limited resources, the maximum size of the training set was limited to 5000 text samples, one training epoch, a batch size of 64, embedding dimension of 32. Despite the reduced efficiency of the LSTM algorithm, the LSTM provided clear evidence referring to potential and capabilities of deep learning models in the field of text classification. (Fig.5)

```
# LSTM
sample_size = 5000
X_train_sample = X_train[:sample_size]
y_train_sample = y_train[:sample_size]
```

Figure 5 *LSTM*

Evaluation

The corresponding models were evaluated according to four performance metrics including accuracy, precision, recall, and F1 score. Accuracy is an overall metric of how frequently predictions were correct, while precision is the proportion of true positives based on positive predictions. On the other hand recall represents the proportion of true positives found from all true positives, and F1 score is the representation of balance between precision and recall. The metrics were calculated using scikit-learn's evaluation measuring functions. Afterwards, all the results gathered were stored as a dictionary, and the results were able to be visualized in a bar graph, making it relatively straightforward to observe all models on the same chart. The evaluation pipeline ensured consistency and equity between models.

The data set used in the current study consisted of 1,583,691 Twitter tweets, about evenly split between sentiment classes (50.1% positive and 49.9% negative), and hence well-suited for objective model testing. A comprehensive data quality check revealed no missing values, an extremely low rate of duplication of about 0.14%, and significant variability in the lengths of tweets (mean = 74.47 characters, std = 36.2), consistent with previous reports that Twitter data is short-lengthed and noisy **[3]**. Preprocessing involved token cleaning and noise reduction, as used in earlier studies to improve performance for social media text in an informal setting **[4]**.

Five classifiers were trained and tested: Logistic Regression, Naive Bayes, Linear Support Vector Machine (SVM), Random Forest, and a deep learning

LSTM-based model. Their performance, in terms of accuracy, precision, recall, and F1 score.(Table 1)

Table 1. *Performance Comparison of Machine Learning Models on Text Classification Task*

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.7586	0.7484	0.7833	0.7654
Naive Bayes	0.7453	0.7436	0.7531	0.7483
Linear SVM	0.7579	0.7463	0.7854	0.7653
Random Forest	0.6773	0.6353	0.8411	0.7238
LSTM	0.6099	0.5826	0.7905	0.6708

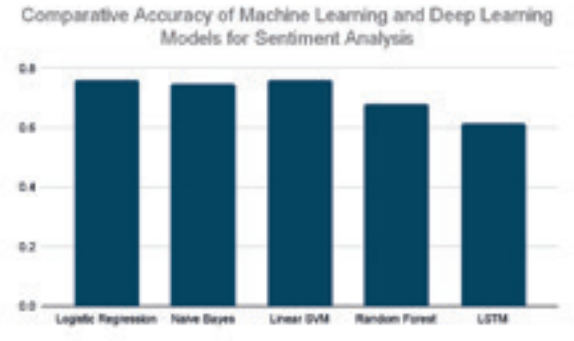

Figure 6: *Comparative Accuracy of Machine Learning and Deep Learning Models for Sentiment Analysis*

The accuracy examination demonstrates that simple machine learning models can be significantly more accurate than deep learning models under controlled conditions. Logistic Regression (75.86%) and Linear SVM (75.79%) are reaching around 75.8% accuracy rate (75.86% for Logistic Regression and 75.79% for Linear SVM) **as shown in Figure 6,** while supporting the effectiveness of the corresponding method when it comes to handling high-dimensional, insufficient data such as tweet texts. This finding is in line with that of **[3],** who highlighted that linear classifier, and SVMs in particular, are optimal for short-text sentiment analysis since they manage to monitor optimum separating hyperplanes in sparse feature spaces. In addition, accuracy results are supported by **[1]** as one of the previous studies as well, while indicating that standard models perform well on Twitter data. It is important to emphasize that Naive Bayes was close behind these results with accuracy equal to 74.53% while Random Forest did well, with 67.73% there was a significant decline in performance. On the other hand, achieving relatively low accuracy of 0.6099 **as shown in Figure 6.** while demonstrating a wide range of limitations when it comes to the interpretation of the LSTM model to the brief texts with minimal context without depending on more complex architectures or pretrained incorporated data. This opposed the estimated outcome of the deep learning model achievements from **[2]**, expanding on the statement that neural networks most commonly require considerable tuning alongside with other semantic resources before overtaking simpler models in the tasks related to the classification of the texts.

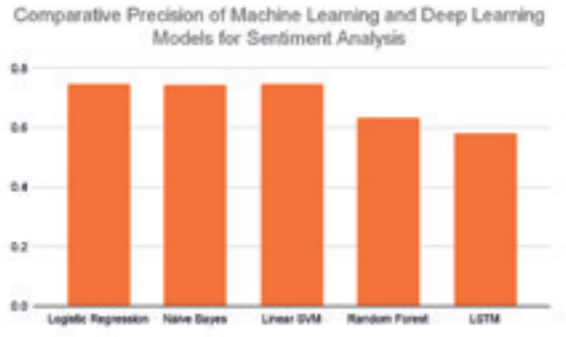

Figure 7: *Comparative Precision of Machine Learning and Deep Learning Models for Sentiment Analysis*

Precision results highlight that the traditional methods are effective for Twitter sentiment analysis. Logistic Regression has the highest precision at 74.8%, followed closely by Linear SVM at 74.6% has a marginally lower precision rate **shown in Figure 7.** within the same variation range, demonstrating identical effectiveness. Taking it into account there is an additional reinforcement of SVM model determined chronology of performance in corresponding NLP tasks, especially in cases when comprehension takes a secondary function prioritizing performance. However, Naive Bayes had 74.4% when it comes to precision rate, as we can see from **Figure 7.** reflecting marginal variation when it comes to overall performance of models which were subject of examination. Following that, results of precision metric are consistent with previous work done by **[1]**, where the development of corresponding model is also reported to be a reliable alternative on emoticon-labeled data similar to the Sentiment140 dataset. It is important to emphasize that Random Forest has reached 63.5%, while LSTM achieved 58.3% precision rate as **shown in Figure 7.** as well, indicating far lower results in the field of positive sentiment identification, revealing

several types of constraints in terms of implementing the LSTM model to brief texts, as mentioned earlier.

One of the reasons for the performance inconsistencies could be that traditional models, such as TF-IDF, are able to perform more effectively with incomplete information. However, the LSTM method developed in the current research involved hyperparameter adjustment, contextual embeddings, and large datasets, while weak performance had been expected given the absence of pretrained embeddings, along with concurrently fragmented and misleading format of tweets.

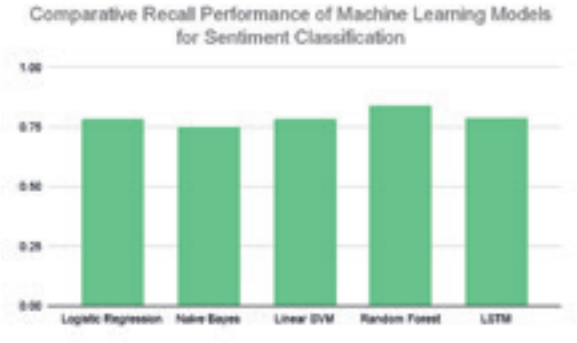

Figure 8: *Comparative Recall Performance of Machine Learning Models for Sentiment Classification*

Recall performance provides insight into the different strengths of the models in identifying true positive sentiments. Random Forest had the highest recall performance with 84.1%, while LSTM achieved 79.1% and Linear SVM with a very close percentage of 78.5% **as is represented in Figure 8.** Logistic Regression has reached 78.3%, although Naive Bayes has the lowest recall percentage among recall performance results which equals to 75.3% **as shown in Figure 8 as well.** The difference between the samples tends to be related to the capacity of various models to identify more positive sentiment instances (in this case positive sentiment), with the more responsive deep learning and ensemble frameworks achieving more efficient performance then more resistant traditional methods. Taking into consideration ensemble structure and capacity to compromise precision in favor of identifying more positives, Random Forest most certainly achieves high levels of recall. Indicating the tendency to overpredict positive sentiment while building up a rate of false positives. Despite the accomplishments of models based on trees on the corresponding textual classification tasks, responsiveness to noise on high-dimensional data was certainly the reason for these results as well, which is in line with **[4]**.

Corresponding results demonstrate a wide range of limitations, when it comes to implementing the LSTM model to brief phrases with minimum context, while it is not depending on more advanced architectures or preconditioned data. This opposed the estimated outcome of the deep learning model achievements from **[2]**, expanding on the statement that neural networks most commonly require considerable tuning alongside with other semantic resources before overtaking simpler models in the tasks related to the classification of the texts. In addition the results of this analysis also align with the findings of **[1]**, since their learning was supported through using the Sentiment140 dataset, which had emoticon labelled tweets, and was based on an extensive repository of labelled data that provided communicative labels for comprehension, which implements recall-based learning.

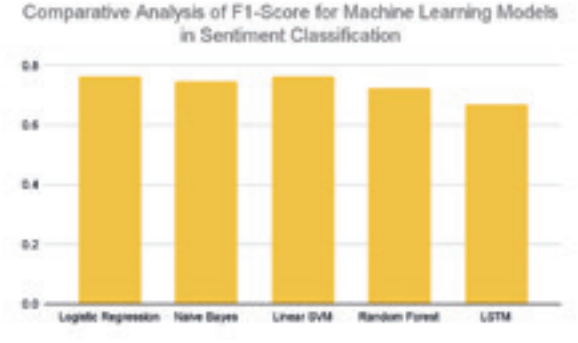

Figure 9: *Comparative Analysis of F1-Score for Machine Learning Models in Sentiment Classification*

Furthermore, in the context of Twitter sentiment analysis, the F1 evaluations indicate the manner in which conventional models perform while maintaining an even balance between precision and recall. The most outstanding values of F1 Score were reported by Logistic Regression and Linear SVM, achieving 76.5% regarding both two models respectively **such as Figure 9. represents**. On the other hand, Naive Bayes followed closely behind with a recall percentage of 74.8% **as shown in Figure 4.** The convenience of Naive Bayes is an important benchmark because of its responsiveness and decent performance, particularly in systems that are running in real time and when computational resources are limited as it is explained by **[1]**. Each of these models discussed above

achieved a balance between evaluating real positives and decreasing false positive results, which is crucial when handling simple terms, as well as fragmented text such as tweets. Random Forest has achieved 72.4%, while LSTM generated a lower percentage of 67.1% when it comes to F1 Score **as it is visible from Figure 9.**, which additionally reflects the reliability of their results, to indicate nothing of the outstanding recall they reported in certain instances. Traditional models perform better in terms of F1 performance, which is in accordance with **[1]**, while indicating that various methods including Logistic Regression consistently performed effectively while trained on the **Sentiment140 dataset**. The results demonstrate that traditional machine learning models still have a competitive advantage, especially in contexts with well-preprocessed and balanced data, as well as reduced tweet lengths. This also corresponds with **[3]** and **[4]**, addressing that Logistic Regression and SVM will outperform deep models in short-text conditions without significant semantic complexity.

The main reason for the corresponding performance was the capacity of conventional models to manage fragmented text representation (TF-IDF). Taking it into consideration Logistic Regression could be fractionally implemented on social media information, since it can be modified to informal language and have reduced capacity in order to prevent overfitting. The limited dataset size could have been responsible for low performance of the LSTM in corresponding research, while being determined in the field of consistent data that reflects social media content and insufficient contextual embeddings. On top of that, low performance results achieved by the LSTM model regardless its ability to process sequential context are in the line with **[1]** and **[5]**, while domain-sensitive limitations, such as acronyms, informal language, and sarcasm, are required to be responded to and solved more effectively, which involves hybrid or attention models.

Future research should be focused on combining contextualized embeddings (BERT, RoBERTa) and investigation of the ensemble methods that provide balance precision and recall. Sentiment misclassifications could also be assisted with comprehension evaluations, uncovering feature importance or using attention mechanisms to detect limited textual indicators.

The corresponding research is important for sentiment analysis because it established a comprehensive benchmark comparing classical machine learning methods - Logistic Regression, Naive Bayes, SVM, and Random Forest - with a deep learning model, LSTM, and provided relevant information related to the differences between accuracy, scalability, and computational effectiveness through an objective evaluation of the five methodologies.

This study operates under real-world conditions, including limited hardware and reduced dataset size (limiting the LSTM training, for instance, to 5000 samples) in order to to make the results more relevant for working in real-time, and potentially useful in implementing related research, opposing several studies that perform in ideal environments with a wide range of resources. Additionally, through evaluating all models using accuracy, precision, recall, and F1 Score metrics, the research presented a balanced, and comprehensive assessment of the models in terms of the advantages and limitations of each model, especially around imbalanced or complex sentiment datasets.

This research presents a responsive and uniform pipeline with an accurate and consistent evaluation of a range of models: whether they be commonly used linear classifiers, decision tree approaches, or recent deep learning architectures such as LSTM. The use of common processing and evaluation methods enhances reproducibility across models, providing an important baseline for future work and extensibility. Furthermore, this work includes an effective LSTM implementation which demonstrates how significant performance can be achieved even with limited computational capacity - an important contradiction that is most frequently ignored in sentiment analysis work. Moreover, this project has a balanced approach towards performance and interpretability. Even though deep learning models provide more contextual information, it is important to emphasize efficiency and transparency for the less complex models, such as Logistic Regression. This work is useful for implementation across different fields where comprehensibility is nevertheless taken into account, even if it is not as critical as accuracy.

CONCLUSION

This study investigated sentiment analysis on social media data, while evaluating the performance of five machine learning models (Logistic Regression, Naive Bayes, Linear SVM, Random Forest, and LSTM),

trained against the Sentiment140 dataset and vectorized using TF-IDF. Research found that out of the five models evaluated, traditional classifiers (Logistic Regression, Linear SVM) produced the most balanced results with F1-scores of 76.5% and 76.5%, respectively, demonstrating that the classical models are more relevant when working with highly limited and dimensional text data that are typical for Twitter. Analysis of precision scores further supported that classical methods were more efficient, while recall scores indicated Random Forest and LSTM performed well when it comes to identification of a wider range related to positive sentiments, even though compromising the precision.

Corresponding research supports findings of **[1]** indicating that distant supervision model and development of a scalable methodology for classifying sentiment while incorporating data labeled using emoticons. Research also supports the outcomes of **[4]** that traditional machine learning models are effective methods for sentiment analysis datasets, particularly because traditional approaches are simple to interpret and require low resources, limiting input and complexity. Furthermore, the restrictions observed in current research with the deep-learning models (LSTM), as discussed by **[5]** , indicated that deep learning models need to be effectively optimized with large, appropriately interpreted datasets, and context sensitive embeddings to successfully outperform their alternatives.

Future research could incorporate contextual word embeddings (eBERT or GloVe) and transformer-based architectures to improve semantic understanding for tweets, preprocessing specific to a domain, hyperparameter tuning, and ensemble methods. There is also potential for extending the application to multilingual sentiment datasets, real-time analysis scenarios, and other useful applications.

The approach used in this paper creates a modular, extensible framework for exploring different approaches to text classification. By allowing for a direct comparison of traditional machine learning techniques and modern deep learning techniques, this approach can suggest practical findings regarding strengths and weaknesses among both categories of techniques. In this study, we were able to optimize both the models and the sampling strategy to make the study feasible, but one advantageous next step would be to increase the dataset size, play with different neural models such as transformer models, along with hyperparameter tuning or cross-validation, to achieve more generalization. This study serves as a basis for future work to create scalable, well-performing text classification systems.

References

[1] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," Stanford University, 2009. [Online]. Available: https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf

[2] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018, doi: 10.1002/widm.1253.

[3] A. Giachanou and F. Crestani, "Like it or not: A survey of Twitter sentiment analysis methods," *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–41, 2016, doi: 10.1145/2938640.

[4] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014, doi: 10.1016/j.asej.2014.04.011.

[5] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014. [Online]. Available: https://arxiv.org/abs/1408.5882

Received: April 28, 2026
Accepted: May 4, 2026

About the authors

Boris Borovčanin is an engineering graduate and MSc student at International Burch University in Sarajevo, Bosnia and Herzegovina. He has academic and practical experience in data analysis and information systems, including work at Raiffeisen Bank dd BIH. His research interests include data science, network security, and machine learning applications.

For citation

Boris Borovčanin, Sentiment Analysis on Social Media Content, *JITA – Journal of Information Technology and Applications, Banja Luka*, Pan-Europien University APEIRON, Banja Luka, Republika Srpska, Bosna i Hercegovina, JITA 16(2026)1:40-47, (UDC: 004.738.5:[658.8:659.1), (DOI: 10.7251/JIT2601040B), Volume 16, Number 1, Banja Luka, June (1-76), ISSN 2232-9625 (print), ISSN 2233-0194 (online), UDC 004