Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 8, S. 372-379, 4, 2024
© Telif hakkı IJANSER'e aittir
Araştırma Makalesi

IJANSER

https://as-proceeding.com/index.php/ijanser
ISSN: 2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 8, pp. 372-379, 4, 2024
Copyright © 2024 IJANSER
Research Article

Sentiment Analysis and Rating Prediction for App Reviews Using Transformer-based Models

Gokberk ESER[1], Cagri SAHIN [2]

[1]*Department of Computer Engineering, Gazi University, Ankara, Turkiye*
[2]*Department of Computer Engineering, Gazi University, Ankara, Turkiye*

[*]*(gokberk.eser1@gazi.edu.tr) Email of the corresponding author*

(Received:16 May 2024, Accepted: 25 May 2024)

(3rd International Conference on Engineering, Natural and Social Sciences ICENSOS 2024, May 16-17, 2024)

ATIF/REFERENCE: Eser, G. & Şahin, Ç. (2024). Sentiment Analysis and Rating Prediction for App Reviews Using Transformer-based Models. *International Journal of Advanced Natural Sciences and Engineering Researches*, 8(4), 372-379.

Abstract – In this study, we present the sentiment analysis of Spotify app reviews, the implementation of Natural Language Processing (NLP) methods, and the use of transformer-based models including BERT, DistilBERT, RoBERTa, and XLM-RoBERTa. Comprehensive preprocessing, including emoji removal, typo correction, and tokenization, was utilized for processing Spotify app reviews from the Google Play Store. Sentiments were analyzed using the VADER Sentiment Intensity Analyzer, categorized into positive, neutral, and negative. Models were assessed for accuracy, precision, recall, and F1-score. DistilBERT achieved the highest accuracy and recall 71.68%, while XLM-RoBERTa demonstrated the best balance with an F1-score of 69.24% in predicting Spotify app ratings.

Keywords – Sentiment Analysis, NLP, User App Review, Transformer Models, Classification, Rating Prediction

I. INTRODUCTION

In the 21st century, mobile technology is considered a significant factor influencing interpersonal relationships, with mobile apps being essential in daily life. Studies have shown that, on average, individuals look at their phones about 144 times a day, spending 88% of this time on various applications [1][2]. The frequency of engagement indicates the significant importance of applications used in contemporary societies and their substantial contribution to the mobile economy, projected to reach a value of $673.8 billion by 2027 [3].

This study uses artificial intelligence to analyze sentiments in a large collection of reviews from the Google Play Store about the Spotify app, which features 601 million monthly active users and approximately 254 million premium subscribers [4]. It leverages advanced Natural Language Processing (NLP) techniques, offering a comprehensive workflow from data gathering and preparation to categorization and analysis.

We utilized transformer-based models such as BERT, DistilBERT, RoBERTa, and XLM-RoBERTa, chosen for their effectiveness in tackling the intricacies of language found in user-generated content. Comprehensive preprocessing techniques, such as emoji removal, typo correction, and tokenization, were used to prepare review data for the analysis of Spotify app reviews from the Google Play Store. Using the

VADER Sentiment Intensity Analyzer, sentiments were classified into positive, neutral, and negative categories based on the expressed sentiments. The models were evaluated on their accuracy, precision, recall, and F1-score. Among the models, DistilBERT showed the highest accuracy and recall, both at 71.68%, while XLM-RoBERTa demonstrated a superior balance between precision and recall with the best F1-score of 69.24% in predicting the ratings of Spotify app reviews. XLM-RoBERTa's performance in predicting ratings for Spotify app user reviews showcases high accuracy in predicting rating 5, but it struggles more with lower ratings 1 and 2, exhibiting a noticeable pattern of misclassifications between adjacent rating categories.

II. RELATED WORKS

Mobile app reviews serve as a critical resource for marketers, researchers and developers, providing comprehensive insights into users' opinions and experiences. These reviews are essential for identifying user satisfaction, app issues, feature requests, and market trends.

Previous studies have utilized diverse approaches to extract insights from user reviews, offering valuable strategic guidance to mobile app developers. For instance, tools like AR-Miner, developed by Chen et al. [5], have contributed to improving application functionality and user experience through the analysis of user feedback.

Panichella et al. [6] conducted a comparative study to investigate the effectiveness of using Natural Language Processing, Sentiment Analysis, and Text Analysis techniques individually or in combination. They focused on extracting features from user comments and introduced an approach to analysis by highlighting the effectiveness of combining these techniques.

Further research by Ali et al. [7] and Noei et al. [8] has explored user experiences across various mobile platforms, integrating these insights into development strategies. Additionally, Pagano and Maalej [9] and Aljrees et al. [10] highlights the importance of mobile application marketing and user satisfaction strategies, offering actionable advice on effectively utilizing feedback from app stores.

More recently, sentiment analysis of mobile app reviews has been utilized to gain additional insights. For instance, Wong et al. [11] employed Multinomial Naïve Bayes and Random Forest algorithms to classify Snapchat app reviews. The implication is that these approaches can be implemented in real-world settings. This study not only classified sentiments into positive, neutral, or negative categories but also evaluated the performance of the classification models using accuracy, precision, recall, and F1-score metrics.

Verma et al. [12] utilized sentiment analysis to explore the nuances of user feedback on advanced language models for ChatGPT, highlighting the critical role of user comments in refining app features. Their findings illustrate how detailed analysis can guide improvements in app functionality and user interface design.

Moreover, in sentiment classification, SentiWordNet 3.0 and Naive Bayes classifiers have been found useful in accurately identifying the sentiment polarity of user comments. The study by Sultana and Sarker [13] also demonstrated the application of fine-grained sentiment analysis for assessing user sentiment towards application features. Guzman and Maalej [14] subsequently evaluated fine-grained sentiment analysis to enhance their recall and accuracy in understanding user sentiment.

III. METHODOLOGY

In this section, we detail our methodology, including dataset collection, data preprocessing, sentiment labeling with VADER, utilization of Transformers Models, and determination of hyperparameters for the models.

A. *Dataset Collection*

The first step of our work was to gather user reviews and ratings for the Spotify app from the Google Play Store. The dataset is stored in a JSON file and includes 500,000 user reviews collected between Nov-21 and Feb-23. It contains 5 variables: date, rating, review, reply review, and reply date.

B. *Data Preprocessing*

In the domain of text-based data analysis, such as machine learning or natural language processing (NLP), preprocessing involves a series of essential steps aimed at preparing raw text data for effective analysis. This preparation phase is crucial for ensuring that the data are cleaned, normalized, and structured, thereby enhancing the quality of the insights derived from subsequent analyses.

In the initial phase of preprocessing text data for analysis, two critical steps are undertaken to ensure the dataset's cleanliness and relevance. To ensure the dataset is clean, we follow the subsequent steps:

- Removing Emojis: We used the emoji library [15] to remove all emojis from the text. This ensures cleaner and more focused text analysis, as emojis cannot be properly processed by most language processing tools, impacting analysis accuracy.
- Filtering Non-English Reviews**:** After removing the emojis, we filtered out non-English reviews by detecting the text's language using the fasttext library [16].
- Typo Correction: In order to enhance data quality and improve result reliability, we aimed to correct spelling mistakes and grammatical problems in texts. To achieve this, we leveraged the language_tool_python library [17] to automatically correct spelling mistakes and grammatical errors. For example, in the sentence "Plz unblock my account", the word "Plz" is corrected to "please", enhancing accuracy and understandability to "Please unblock my account".

After completing these initial cleaning steps, we then proceed with standard preprocessing techniques to further prepare the text.

- Tokenization: We first tokenized the text, splitting it into individual words or tokens. This process is fundamental for analyzing the text at the word level and preparing it for further preprocessing steps. Tokenization allows us to apply more sophisticated NLP techniques to a structured form of the text.
- Lowercasing: We converted the entire text to lowercase, ensuring words like "House" and "house" are treated identically, thereby creating consistency within the analysis. This step is fundamental in normalizing the dataset for further processing.
- Removing punctuation marks: We eliminated all punctuation marks (e.g., ., ;, :, !, ?) and numbers, whether written as digits or words, to reduce complexity in the text. This simplification helps to streamline the analysis, keeping the emphasis on the text's linguistic and thematic elements.
- Lemmatization: We employ a spaCy [18] language model for lemmatization, simplifying words to their base or root form. As an illustration, various forms of the word "make," including "making," "made," and "makes," are lemmatized to "make." This technique supports more uniform analysis by standardizing diverse manifestations of a word.
- Removing stop-words: We filter out words that provide minimal semantic contribution or are overly common, such as "and," "the," and "but," using a stop words list from the NLTK library [19]. The removal of stop words allows for more meaningful and concentrated analyses of the text, preventing these frequently occurring words from obscuring the analytical results.

Table 1. Preprocessing for the Example Review

Step	Words
Tokenization	[Downloaded, it, and, loved, it, !, So, intuitive, and, quick, support, ., Fixed, my, issue, fast, ., Highly, recommend, !]
Lowercasing	[downloaded, it, and, loved, it, !, so, intuitive, and, quick, support, ., fixed, my, issue, fast, ., highly, recommend, !]
Removing Punctuation	[downloaded, it, and, loved, it, so, intuitive, and, quick, support, fixed, my, issue, fast, highly, recommend]
Lemmatization	[download, it, and, love, it, so, intuitive, and, quick, support, fix, my, issue, fast, highly, recommend]

Removing Stop Words	[download, love, intuitive, quick, support, fix, issue, fast, highly, recommend]
Final Review	download love intuitive quick support fix issue fast highly recommend

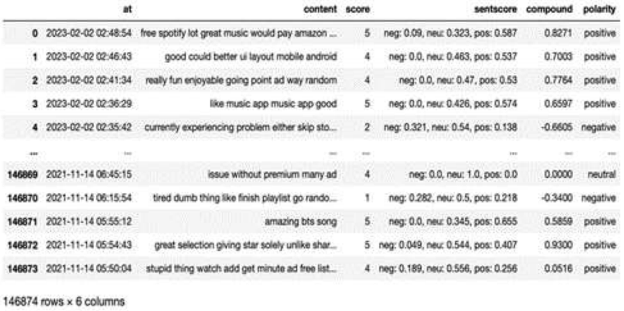
Table 1 illustrates an example of the preprocessing steps we followed. The example review "Downloaded it and loved it! So intuitive and quick support. Fixed my issue fast. Highly recommend!" has been preprocessed, resulting in a cleaned and standardized format for further analysis, represented as "download love intuitive quick support fix issue fast highly recommend."

C. *Sentiment Labeling with VADER*

We used the VADER (Valence Aware Dictionary and sEntiment Reasoner) Sentiment Intensity Analyzer for our analysis. This tool is specifically designed for social media texts, effectively creating and validating a sentiment lexicon tailored to the nuances of microblog-like contexts [20]. Its capability to handle informal language, emoticons, slang, and acronyms makes it particularly suitable for analyzing online reviews.

The sentiment compound score, which represents the overall sentiment intensity of a text, was derived from each review using VADER. This tool categorizes sentiments into positive, neutral, or negative based on whether the compound score is greater than (0.05) for positive sentiment, between (-0.05) and (0.05) for neutral sentiment, or less than (-0.05) for negative sentiment.

Reviews were further scrutinized to eliminate inconsistencies between the expressed sentiment and the user's rating. Reviews rated below 3 but classified as positive, or above 3 but classified as negative, were discarded. Additionally, any neutral reviews with extreme ratings of either 1 or 5 were also excluded. This selective filtering ensures that the analysis considers only reviews whose sentiments and ratings are congruent.



	at	content	score	sentscore	compound	polarity
0	2023-02-02 02:48:54	free spotify lot great music would pay amazon ...	5	neg: 0.09, neu: 0.323, pos: 0.587	0.8271	positive
1	2023-02-02 02:46:43	good could better ui layout mobile android	4	neg: 0.0, neu: 0.463, pos: 0.537	0.7003	positive
2	2023-02-02 02:41:34	really fun enjoyable going point ad way random	4	neg: 0.0, neu: 0.47, pos: 0.53	0.7764	positive
3	2023-02-02 02:36:29	like music app music app good	5	neg: 0.0, neu: 0.426, pos: 0.574	0.6597	positive
4	2023-02-02 02:35:42	currently experiencing problem either skip sto...	2	neg: 0.321, neu: 0.54, pos: 0.138	-0.6605	negative
...	...	...	...	...	...	...
146869	2021-11-14 06:45:15	issue without premium many ad	4	neg: 0.0, neu: 1.0, pos: 0.0	0.0000	neutral
146870	2021-11-14 06:15:54	tired dumb thing like finish playlist go rando...	1	neg: 0.282, neu: 0.5, pos: 0.218	-0.3400	negative
146871	2021-11-14 05:55:12	amazing bts song	5	neg: 0.0, neu: 0.345, pos: 0.655	0.5859	positive
146872	2021-11-14 05:54:43	great selection giving star solely unlike shar...	5	neg: 0.049, neu: 0.544, pos: 0.407	0.9300	positive
146873	2021-11-14 05:50:04	stupid thing watch add get minute ad free list...	4	neg: 0.189, neu: 0.556, pos: 0.256	0.0516	positive

146874 rows × 6 columns

Fig. 1 Compound score and sentiment expression for reviews

Figure 1 illustrates the compound score and sentiment expression for reviews with the given user ratings after preprocessing steps.

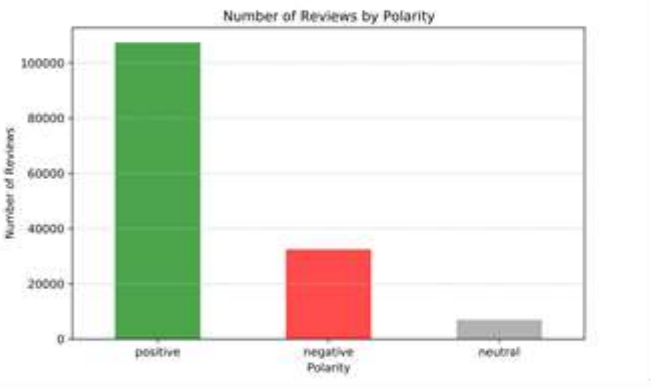


Fig. 2 Number of reviews by polarity

Figure 2 shows the distribution of user reviews by their polarity after preprocessing steps and the elimination of inconsistencies between the expressed sentiment and the user's rating. From 500,000 Spotify app reviews, there were 107,358 positive, 32,511 negative, and 7005 neutral polarity reviews identified.

D. *Transformers Models*

In our study, we compare 4 transformers model which are BERT, DistilBERT, RoBERTa, and XLM-RoBERTa.

BERT, a powerful language representation model developed by Google, is designed for pre-training deep bidirectional representations from unlabeled text. It achieves this by considering both left and right context at all layers [21]. The training of the BERT model occurs in two stages: preliminary training and fine-tuning. During preliminary training, the model learns the language structure by attempting to predict randomly masked words in a large unlabeled text dataset. In the second stage, it is fine-tuned for specific NLP tasks such as sentiment analysis, text summarization, and question-answering, customizing the output layer of the model to achieve effective results. The BERT model used in this study was originally trained on the Book Corpus and English Wikipedia. This training enabled it to learn language structures across diverse contexts and adapt to various natural language processing (NLP) tasks.

DistilBERT, designed as a more efficient version, provides a lightweight alternative to the traditional BERT architecture [22]. It uses an advanced distillation technique to remove some layers and reduce the remaining layers by 40%, enhancing processing speed and efficiency while retaining 97% of the language understanding capability. This allows the model to train faster and produce outputs more quickly.

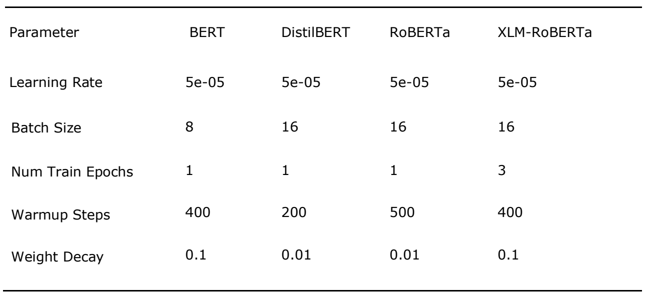
RoBERTa, an advanced optimization of the BERT model introduced by Facebook AI in 2019 [23], is designed for deeper language structure learning. Trained on larger datasets for extended periods using dynamic masking techniques instead of static ones, it focuses solely on the Masked Language Modeling (MLM) task. This approach allows different masked versions in each training instance, enhancing language learning comprehensiveness

XLM-RoBERTa, short for Cross-Lingual Language Model-Robust Optimized Bidirectional Encoder Representations from Transformers, is trained on the extensive 2.5 TB CommonCrawl dataset in 100 languages. It learns language structure through the Masked Language Modeling (MLM) method [24].

E. *Hyperparameters For Models*

Optimizing hyperparameters is crucial for maximizing the performance of machine learning models, directly influencing their learning capacity and generalization abilities. To identify optimal parameters for each model, we utilized the Optuna library for hyperparameter search. Optuna enhances optimization processes by enabling the creation of customized search spaces [25]. Using the framework provided by Optuna, we searched for optimal hyperparameters for considered transformer language models.

Table 2. Optimal Hyperparameters for Transformers Model



Parameter	BERT	DistilBERT	RoBERTa	XLM-RoBERTa
Learning Rate	5e-05	5e-05	5e-05	5e-05
Batch Size	8	16	16	16
Num Train Epochs	1	1	1	3
Warmup Steps	400	200	500	400
Weight Decay	0.1	0.01	0.01	0.1

Table 2 presents the optimal hyperparameters, such as learning rate, batch size, number of training epochs, warm-up steps, and weight decay, for each transformer model.

F. *Performance Evaluation Metrics*

In performance evaluation metrics, "true positive," "true negative," "false positive," and "false negative" are represented by "TP," "TN," "FP," and "FN" respectively. Accuracy measures the proportion of correct predictions made by the model and is represented as:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

Precision quantifies the accuracy of positive predictions, which is crucial when false positives have significant impacts. Its formula is:

$$Precision = TP / (TP + FP) \tag{2}$$

Recall, calculated as the proportion of true positive predictions in a dataset to all actual positives, is a key metric in performance evaluation.

$$Recall = TP / (TP + FN) \tag{3}$$

The F1-score is calculated as the harmonic mean of precision and recall, combining their values into a single metric that balances accuracy and completeness.

$$F1\,Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{4}$$

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In our experiments, we used a Google Colab Tesla V100 GPU and implemented the Python-based Huggingface-transformers library [26]. We evaluated the performance of the considered transformer-based models on sentiment analysis of a dataset comprising 146,874 Spotify app reviews. All models were trained on the designated training data, and their performance was subsequently assessed on the test data. The dataset was partitioned into training, validation, and test sets at ratios of 60%, 20%, and 20%, respectively. This configuration involved utilizing 88,124 data points for training, 29,375 for validation, and 29,375 for testing, extracted from the total dataset.

Table 4. Performance Metrics for Transformer Models

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
BERT	71.54%	68.22%	71.54%	68.93%
DistilBERT	71.68%	68.31%	71.68%	68.91%
RoBERTa	71.02%	67.76%	71.02%	68.51%
XLM-RoBERTa	71.34%	68.53%	71.34%	69.24%

The data presented in the Table 4 suggests that among the four compared transformer models, they all demonstrate closely matched performance. The DistilBERT model stands out prominently for achieving the highest Accuracy and Recall, reaching 71.68%. In contrast, XLM-RoBERTa demonstrates the best performance in terms of F1-Score, achieving 69.24%.

The confusion matrix in Table 5 illustrates XLM-RoBERTa's performance in predicting ratings for Spotify app user reviews. The rows represent the predicted ratings by the model 1 to 5, while the columns represent the actual ratings 1 to 5. For instance, in the cell corresponding to "Predicted (1)" and "Actual (1)," the value 4160 indicates that the model predicted 4160 instances as rating 1, and these instances were indeed rated as 1 in reality.

Table 5. Confusion Matrix for the XLM-RoBERTa Model

Rating	**Actual (1)**	**Actual (2)**	**Actual (3)**	**Actual (4)**	**Actual (5)**
Predicted (1)	4160	69	215	6	29
Predicted (2)	958	229	322	67	12
Predicted (3)	662	215	1767	840	546
Predicted (4)	37	89	891	1680	1994
Predicted (5)	41	3	427	997	13119

Similarly, in the cell corresponding to "Predicted (2)" and "Actual (3)," the value 322 indicates that the model predicted 322 instances as rating 2, but these instances were actually rated as 3. The model demonstrates exceptional performance in predicting the highest rating, accurately identifying 13,119 out of 14,556 actual rating 5 reviews. This showcases its strong capacity for recognizing features of highly positive feedback. However, The model struggles more with lower ratings such as1 and 2. There's also a noticeable pattern of misclassifications between adjacent rating categories. For instance, many instances rated as 3 are incorrectly classified as either 2 or 4.

V. Conclusion

Overall, this study has effectively categorized user reviews into three sentiment categories: positive, neutral, and negative, utilizing the VADER library. Following the determination of review polarity, transformer-based models such as BERT, DistilBERT, RoBERTa, and XLM-RoBERTa were applied to predict user ratings.

The experimental results revealed that DistilBERT performed slightly better than the others in terms of Accuracy and Recall. Additionally, XLM-RoBERTa achieved the best F1 Score and exhibited consistent classification performance by maintaining a balance between Precision and Recall. XLM-RoBERTa performs well in predicting high ratings, particularly rating 5, but it encounters challenges with lower ratings, especially 1 and 2, showing a distinct pattern of misclassifications between adjacent rating categories.

For future studies, we plan to explore additional classification algorithms and investigate alternative preprocessing methods and classification techniques to potentially enhance performance outcomes. Moreover, utilizing a larger dataset for classification might lead to more effective models.

References

[1] Kerai, A. (2024). Cell Phone Usage Statistics: Mornings Are for Notifications. Reviews.org. Retrieved March 20, 2024, from https://www.reviews.org/mobile/cell-phone-addiction/,

[2] Wurmser, Y. (2020). The Majority of Americans’ Mobile Time Spent Takes Place in Apps. EMARKETER. Retrieved March 21, 2024, from https://www.emarketer.com/content/the-majority-of-americans-mobile-time-spent-takes-place-in-apps

[3] Mobile app revenue worldwide 2019-2027, by segment. (2023). Statista. Retrieved March 22, 2024, from https://www.statista.com/forecasts/1262892/mobile-app-revenue-worldwide-by-segment

[4] Turner, A. (2024). Spotify Users: How Many People Have Spotify? . BankMyCell. Retrieved March 26, 2024, from https://www.bankmycell.com/blog/number-of-spotify-users/

[5] Chen, N., Lin, J., Hoi, S. C. H., Xiao, X., & Zhang, B. (2014). AR-miner: mining informative reviews for developers from mobile app marketplace. Proceedings of the 36th International Conference on Software Engineering. https://doi.org/10.1145/2568225.2568263

[6] Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G., & Gall, H. C. (2015). How can i improve my app? Classifying user reviews for software maintenance and evolution. 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME). https://doi.org/10.1109/icsm.2015.7332474

[7] Ali, M., Joorabchi, M. E., & Mesbah, A. (2017,). Same App, Different App Stores: A Comparative Study. 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft). https://doi.org/10.1109/mobilesoft.2017.3

[8] Noei, E., Zhang, F., & Zou, Y. (2021). Too Many User-Reviews! What Should App Developers Look at First? IEEE Transactions on Software Engineering, 47(2), 367–378. https://doi.org/10.1109/tse.2019.2893171

[9] Pagano, D., & Maalej, W. (2013). User feedback in the appstore: An empirical study. 2013 21st IEEE International Requirements Engineering Conference (RE). https://doi.org/10.1109/re.2013.6636712

[10] Aljrees, T., Umer, M., Saidani, O., Almuqren, L., Ishaq, A., Alsubai, S., Eshmawi, A. A., & Ashraf, I. (2024). Contradiction in text review and apps rating: prediction using textual features and transfer learning. PeerJ Computer Science, 10, e1722. https://doi.org/10.7717/peerj-cs.1722

[11] Wong, W. H., Ismail, S., Arifin, M. A., Make, S. S. A., Wahab, M. H. A., & Shaharudin, S. M. (2021). Sentiment Analysis of Snapchat Application’s Reviews. 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS). https://doi.org/10.1109/aidas53897.2021.9574379

[12] Verma, P., Srivastava, R., Fatima, S., & Pratap, A. (2024). Sentiment Analysis on ChatGPT Reviews. 2024 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS). https://doi.org/10.1109/sceecs61402.2024.10482121

[13] Sultana, R., & Sarker, S. (2018). App Review Mining and Summarization. International Journal of Computer Applications, 179(38), 45–52. https://doi.org/10.5120/ijca2018916918

[14] Guzman, E., & Maalej, W. (2014). How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews. 2014 IEEE 22nd International Requirements Engineering Conference (RE). https://doi.org/10.1109/re.2014.6912257

[15] emoji — emoji documentation. [Online]. Available: https://carpedm20.github.io/emoji/docs/

[16] fastText. [Online]. Available: https://fasttext.cc/

[17] LanguageTool. [Online]. Available: https://github.com/jxmorris12/language_tool_python

[18] spaCy · Industrial-strength Natural Language Processing in Python. [Online]. Available: https://spacy.io/

[19] NLTK :: Natural Language Toolkit. [Online]. Available: https://www.nltk.org/

[20] Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. Proceedings of the International AAAI Conference on Web and Social Media, 8(1), 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

[21] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[22] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[23] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[24] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.

[25] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3292500.3330701

[26] Transformers. [Online]. Available: https://huggingface.co/docs/transformers/index