Project Writeup

Sentiment Analysis and Topic Modeling on Threads App Reviews

I. Business Problem

Since Elon Musk acquired Twitter in April 2022 (Siddiqui & Gregg, 2022), his decision to monetize the application programming interface (API) and several other policy changes caused much turbulence, with many users deciding to leave the platform (Siddiqui et al., 2023). Taking advantage of this business opportunity, Meta launched a text-based social media app called Threads, closely related to Instagram, to compete directly with Twitter. Despite being the fastest-growing consumer software application in history, gaining over 100 million users in its first five days (Ray, 2023), whether Threads can remain competitive in the long run remains an unanswered question. The strategy to channel Instagram users to Threads through a close connection with Instagram failed, as Threads only had 8 million daily active users by the end of the first month since its launch (Rivera, 2023). The downward-spiraling population of active users signaled the strong need for immense creativity and innovation to keep the app relevant to users in the long run.

Most importantly, Threads customer reviews should be analyzed carefully for upcoming updates to fix customers' inconveniences and meet their demands. Therefore, our team wants to train a machine-learning model using the Convolutional Neural Networks (CNN) model to classify the feedback into positive, negative, and neutral through not just the numerical rating, but through the words within the text-based feedback. We would also want to identify the reasons behind this classification through the Latent Dirichlet Allocation (LDA) model to identify possible future improvement areas.

II. Data Preparation

The dataset we employed contains over 37,000 text-based reviews of the New Thread mobile application sourced from both the Google Play Store and Apple App Store (Shuv, 2023). Moreover, reviews are tagged with accompanying star ratings (e.g., 1, 2, 5). This makes it easier to train sentiment analysis models. The original data includes 12 variables, as depicted in Table 1:

No.	Variable name	Type of data	Value
1	source	string	Google Play or App Store
2	review_id	string	id generated after each review done by users
3	user_name	string	Thread usernames by the people who wrote reviews
4	review_title	string	title about review (2000 unique values, accounting for 5% only)
5	review_description	string	main review text
6	rating	integer	corresponding ratings
7	thumbs_up	integer	validation provided by other users
8	review_date	string	date each review was published
9	developer_response	string	100% no response
10	language_code	string	the corresponding language that the review was written in

11	appVersion	string	corresponding version for reviews
12	country_code	string	1 value only: “us”

Table 1: Description of the original dataset.

We decided to drop these features 'user_name', ‘review_id’, ‘developer_reponse’, and ‘country_code’ because they neither have any data nor show a direct impact on the ratings and reviews.
The remaining features involve:

+ review_description: By analyzing the words and phrases used, wecan understand the overall sentiment of the user's experience. This also provides insights into areas for Thread’s improvement.
+ rating: This provides a clear, numerical indication of the user's satisfaction and can be used to train other datasets to recognize the tone of users without specific ratings.
+ thumbs_up: The number of likes a review receives can be an indicator of how much others resonate with the sentiment expressed. High likes for a positive review strengthen the positive sentiment, while high likes for a negative review might suggest a widespread issue.
+ appVersion: This can help identify if sentiment is tied to a specific version of the app.
+ review_date: This allows for sentiment tracking over time.
+ language_code: This allows us to sort out reviews in other languages besides English.

III. Model description

Convolutional Neural Networks (CNNs) are effective tools for sentiment analysis in text classification, valuable for businesses looking to understand customer opinions and feedback. CNNs work by scanning a sentence matrix, which represents the text data, with filters applied to both distributed and discrete word embeddings to create lower-dimensional representations (Amin & Nadeem, 2018), in order to capture both local patterns like specific word sequences indicative of sentiment and global context (Liao et al., 2017). This model helps increase the amount of accurate insights into customer sentiments, enabling better decision-making, targeted marketing strategies, and improved customer satisfaction.

Reason for choosing CNN

- Efficiency of 1D Convolutional filters in CNNs for text classification: CNN-based models for text classification leverage one-dimensional convolutional filters that slide over word embeddings to extract n-gram features (Soni et al., 2022), effectively capturing local patterns while maintaining model simplicity. This approach offers efficiency, as CNNs can process text in parallel and are less resource-intensive than RNNs, LSTMs, and other transformers.
- Automated hierarchical representation learning: Unlike traditional machine learning models, which rely on hand-crafted features, CNNs automatically learn hierarchical representations. Their position invariance and ability to recognize patterns regardless of their location in the text could enhance model’s robustness.
- Ability to prevent overfitting: CNNs stand out in preventing overfitting due to their parameter sharing and regularization techniques. Compared to other models like traditional machine learning algorithms, recurrent neural networks (RNNs), and transformers, CNNs offer efficient learning through data augmentation, pooling layers, and transfer learning, enabling them to generalize better to unseen data and outperform other approaches in mitigating overfitting.

Input

A CNN for text classification takes a matrix, which is a numerical value, as input. To be specific, the text, which in this case is a customer’s review of Threads, is pre-processed and then transformed into a grid (matrix), where each word is represented by a row, and each column corresponds to a feature of the word, which could be demonstrated by an example in Figure 1. This numerical representation allows the CNN to process the text data effectively later on.

Sentence: I love using Threads

	Feature 1	Feature 2	Feature 3
I	1.0	0.0	0.0
love	0.0	1.0	0.0
using	0.0	0.0	1.0
Threads	0.5	0.5	0.0

Figure 1: Example of how text is transformed into a matrix to serve as the input of CNN.

How CNN makes predictions: The order of layers in a CNN model could be depicted in Figure 2 below:

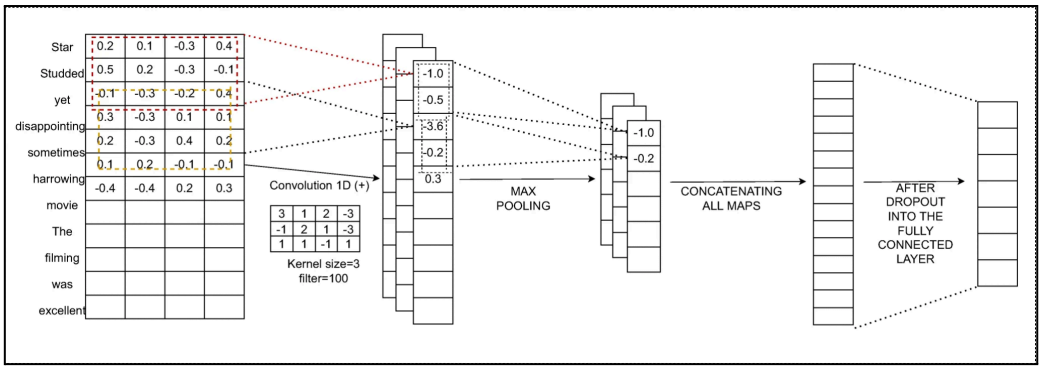

Figure 2: The order of layers in a CNN (Source: Soni et al., 2022).

1. Convolution & Feature Extraction: The "Conv1D" layer scans the sequence of word embeddings looking for patterns related to sentiment.
2. Dimensionality Reduction: The "MaxPooling1D" layer reduces the complexity of the data by selecting the most important features from the previous layer.
3. Data Reshaping: The "Flatten" layer transforms the 3D tensor (which usually contains information about filters, features, and sequence length) into a 1D vector suitable for the final classification layer.
4. Sentiment Classification: The “Fully Connected" layer assigns a probability score to each sentiment class (positive, negative, neutral) using the Softmax activation function. This ensures the probabilities sum to 100%, allowing multi-class classification (positive/ neutral/ negative).

Output

In sentiment analysis using a Convolutional Neural Network (CNN), the output probability distribution across sentiment classes, including positive, negative, and neutral, signifies the model's confidence in each sentiment category. The probability assigned to the positive class indicates the likelihood of the input text expressing a positive sentiment, while the probability for the negative class represents the likelihood of a negative sentiment. Likewise, the probability for the neutral class signifies the model's confidence in the input text being neutral in sentiment.

For example, we have these output probabilities:

- Positive: 0.50
- Negative: 0.40
- Neutral: 0.10

The CNN assigns a 50% probability (0.5) to the text expressing a Positive sentiment. This is the highest score, indicating the model is most confident in the Positive classification. Negative and Neutral classes have lower probabilities, suggesting that while there is some evidence for these sentiments, the model is less certain. Therefore, in this example, it is most likely that the input text data expresses a Positive sentiment.

By examining these probabilities, we can determine the sentiment conveyed in the input text, with thresholds or the class with the highest probability guiding the classification into positive, negative, or neutral categories.

IV. Implementation

- **Data Preprocessing:** In this step, we ensure the review text data is preprocessed and transformed into a numerical format, which fosters consistency, focus, and accuracy. We undertook the seven steps as below:

- Tokenization: We made sure the reviews are split into individual words (tokens).
- Lowercasing: We converted all words to lowercase.
- Remove punctuation and stop words: We removed all punctuation marks and common stop words (such as "a", "an", and "but").
- Remove non-English text: We only kept English text for the model to easily perform sentiment analysis and topic modeling.
- Word Indexing: The words are then converted to numerical indices using a tokenizer.
- Word embeddings: Word indexes are then converted to a dense numerical vector representation (CNN will randomly assign numerical vectors for each word in its vocabulary. These initial vectors typically have decimal values).
- Sequence padding: Sequences are padded to ensure a uniform length, using the embedding vectors, not the original indices.

This final matrix serves as the input for the CNN for our sentiment classification task. The CNN can then analyze the relationships between these word embeddings and their context to understand the sentiment or meaning of the sentence.

- **Exploratory Data Analysis (EDA):** EDA is done by sketching various diagrams to see the distribution and characteristics of the dataset (Shuv, 2023).

1. Bar Plot of Distribution of Review Rating

As depicted in Figure 3, There is a higher number of positive reviews (scores 4 and 5) than negative reviews (scores 1 and 2). This poses an inherent bias in the dataset, in which the model would be better at identifying positive words than negative words.

Meanwhile, the number of review scores of 1 (lowest) and 5 (highest) is approximately 2-3 times the number of review scores of 2, 3, and 4. As most of these feedbacks show strong negative and positive emotions, this contrast makes it easier to train the model to identify positive/negative words.

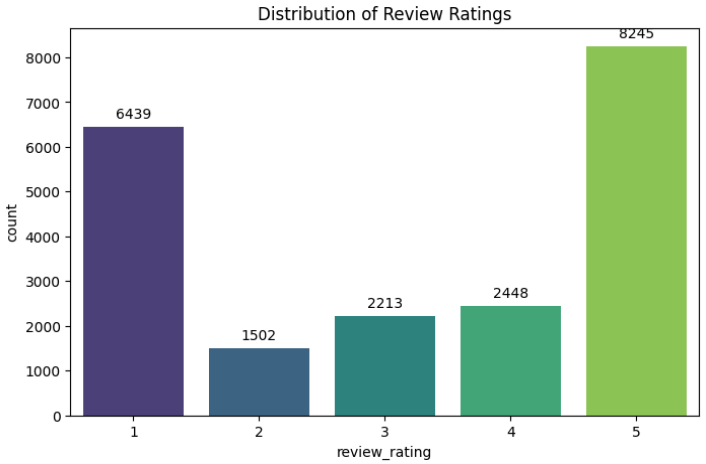

Figure 3: The distribution of Threads' review ratings.

2. Bar Plot of Distribution of Sentiment Labels

Another way to illustrate the data distribution is to classify the scores into sentiment labels. In this case, our group classified scores 1 and 2 as "negative", scores 4 and 5 as "positive", and score 3 as "neutral", as shown in Figure 4. The small number of 3 "neutral" scores is observed across both bar graphs, highlighting the challenges in training the model to identify "moderate" feedback. Nevertheless, this is understandable and acceptable in all review datasets, as people who lean towards a positive or negative experience of the app are more likely to voice their feedback back to

the company, while people who have neutral feelings are more likely to keep silent.

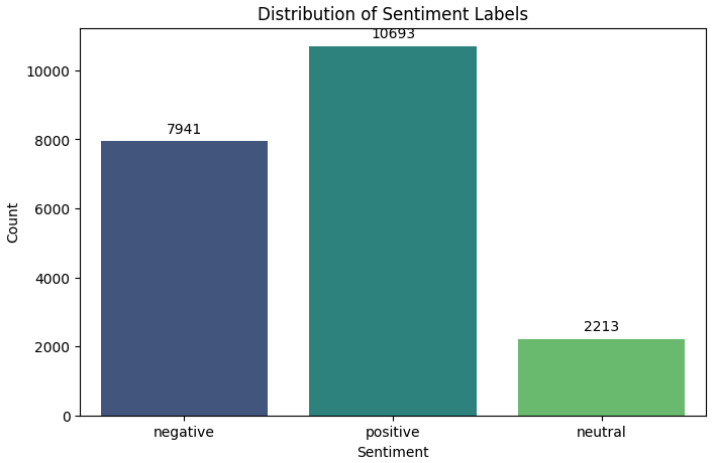

Figure 4: Distribution of sentiment labels.

3. **Identify the Most Common Words**
   As shown in Figure 5, the top most common words in the dataset are mostly positive words, for example, “good”, “like”, “great”, and “nice”, showing a generally more positive experience of the app among Thread users. However, there are still areas for improvement, showing through words like “better” (the app can be updated to become better next time), “can’t” (users are unable to use certain app functions), or “don’t” (suggest a negative experience).

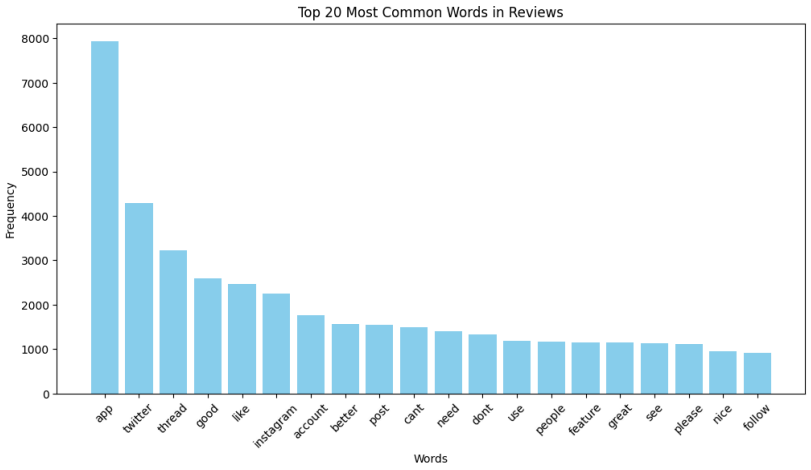

Figure 5: Top 20 most common words in reviews.

- **Model training:** In our neural network, we employed five layers, with a total of 675747 trainable parameters. The first layer, Embedding, plays a key role in transforming the words within each review text into dense vectors of numbers, aiming to capture the semantic meaning between words in the sentence’s context. Next, the Conv1D layer applies the Rectified Linear Unit (ReLU) activation function to capture non-linear patterns in the data, scanning the sequence of word embeddings to identify patterns related to sentiment. Then, a MaxPooling1D layer is constructed to decrease the dimensionality of the data from the previous layer by removing unimportant features. Subsequently, the fourth layer ‘Flatten’ is responsible for flattening the 3D tensor to a 1D vector, which is crucial for sending the data into the final layer. Lastly, the Dense layer makes the final sentiment classification output by assigning a probability score to each positive, negative, and neutral sentiment class. The softmax activation function is employed in this step to ensure the probabilities add up to 100%, which is suitable for multi-class classification for this problem. It is also noted that due to class imbalance, which we discovered during the EDA step, we decided to modify the class weights in the loss function, with the weight of the ‘neutral’ class 5 times higher and the weight of the

'negative' class 3 times higher than the 'positive' one. This ensures our model pays close attention to correctly classifying cases with minority classes.

- **Model cross-validation:** Lastly, to assess our model's robustness, we employed a Stratified K-fold to divide the data into 5 folds, with the ratio of each class being preserved for fair assessment. In each fold, we applied the same model with similar weight distribution in the loss function to calculate the accuracies of each fold as well as the mean cross-validation accuracy overall.

V. Result Analysis

5.1 Sentiment analysis using CNN

- **Test Performance:**

We benchmark our result with the result of a dummy classifier, which always predicts the most frequent class (or 'positive' class in this case). Given the distribution of review sentiments in section IV, we will calculate the dummy classifier's accuracy as the ratio between the number of positive reviews versus the total number of reviews, equivalent to 51,2%. So, the overall test accuracy as of the time of writing this report is 73.1%, which is much higher than the dummy classifier's result. This indicates that our model is much better at predicting the correct sentiment of reviews than a model with random, naive predictions. This also suggests that our model has learned meaningful patterns in the data and can capture some variance of it.

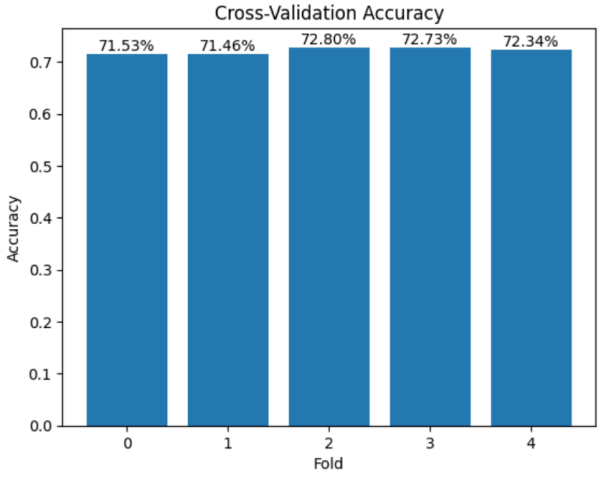

Figure 6. Cross-Validation Accuracy.

Our group also performs Stratified K-fold cross-validation to provide insights into my model's performance consistency across different data subsets. The cross-validation accuracy results are shown in Figure 6. As can be seen, the cross-validated accuracies are relatively close to each other, ranging from approximately 71.46% to 72.8%, as of the time we ran the code to write this report. This indicates that the model performs consistently across different subsets of the data. The small variance in the accuracy values suggests that the model is reliable and not overly sensitive to the specific data in each fold. Furthermore, the mean cross-validated accuracy (approximately 72.1%) is very close to the accuracy on the test set (73.2%). This alignment indicates that our model is stable and generalizes well across new, unseen data without overfitting. Overall, these results prove that our model is robust and performs consistently, offering reliable sentiment predictions across different data subsets.

- **Class-wise Performance Metrics**

Class	Precision	Recall
Negative	0.68	0.83
Neutral	0.33	0.18
Positive	0.82	0.76

Table 2: Model's performance metrics across three classes.

Firstly, from the results shown in Table 2, it can be seen that our model performs notably well in identifying positive reviews, with high precision and recall, meaning that most positive reviews are correctly identified with few false positives. Similarly, the model is relatively good at identifying negative reviews (with notably high recall) but less precise. This means that there are some false positives among the predicted negative reviews. Conversely, the model struggles significantly with neutral reviews, with both low precision and recall of only 33% and 18%. This indicates difficulty in distinguishing neutral reviews from positive and negative ones. We have dived deeper into the top words with the highest weight among positive and negative reviews and found that they typically contain keywords that show extreme feelings such as “poor”, “horrible”, “irrelevant”, “copycat”, “pointless”, and “excellent”. We believe this might be the reason why positive and negative reviews are easily spotted by the model, as compared to unclear, vague keywords in neutral reviews.

5.2 Topic modeling using the LDA model

For positive reviews: To further understand what specific aspects of the Threads application users are interested in, we have performed topic modeling using the LDA technique to extract the top five most common topics among positive results. As depicted in Table 3, we discovered that users are particularly satisfied with the app’s interface, describing it as “lovely” and “fun”, which suggests that they may find it visually appealing and engaging. Additionally, the mention of “Twitter” in both topic 1 and topic 2 indicates that users perceive Threads as a superior alternative to other social media platforms, especially its rival Twitter. Some users also express good experiences with the app's community and interactions with words like "friendly," "like Instagram," and "user”, “wow", implying that they love the app's social features.

Topic	Content
1	lovely interface fun beat twitter good wonderful app one amazing
2	think app way alternative thread much review first better twitter
3	follow like option please feature easy use add need thread
4	im friendly like instagram good thread user wow app top
5	instagram smooth like thread really experience best nice app good

Table 3. Most common topics from positive reviews.

For negative reviews: Conversely, by looking at top 10 common topics from negative reviews, our group identifies that users mostly express frustration with the app's content moderation and community standards, as can be seen in Table 4. Terms like "racist opinion", "truth" and "boring speech" suggest that users feel the app does not effectively manage harmful content or facilitate meaningful discussions. Additionally, the mention of "waste" and "support" implies that users might be dissatisfied with the app's customer service. In Topic 4, users express issues with account management and login processes. Phrases such as "without login," "can't delete," and "Instagram account" reveal that users face difficulties with logging into the app and managing their accounts, including integration with Instagram. This indicates significant frustration with the app's accessibility and user control over their accounts.

Topic	Content
1	racist opinion time truth waste support free boring speech freedom
2	better new clone work app doesnt twitter nothing properly working
3	app search poor following dont see feed post people follow
4	want without login app dont thread cant delete instagram account
5	worse don't suck version best good paste app twitter copy

Table 4. Most common topics from negative reviews.

To sum up, we discovered that from the investigated set of more than 20,000 customer reviews, the Threads app at the time received mixed opinions. In positive reviews, users are satisfied with the well-designed interface, ease of use, and a promising alternative to Twitter and Instagram. However, negative reviews highlight critical issues with content moderation, technical performance, usability, and originality. We believe that it is important for the Threads app to address these negative aspects promptly, particularly in content management and technical stability, which could then significantly improve user satisfaction and overall app success.

VI. Scaling Feasibility

6.1 Business Insights

Our suggested model can be brought to great use in the real business scene for social media applications. By classifying the customer reviews into different tiers, and identifying the keywords each class entails, our sentiment analysis model can help monitor the brand and application's general customer perception, as well as pinpoint the positive and negative touchpoints for customer experience. From such insights, the brand can adjust its product's feature offerings and improve its brand image in general.

While our proposed model only uses primary reviews directly extracted from app stores, future practical development could utilize extracted keywords linked to a sentiment to build follow-up surveys to gain more access to population perception. On those grounds, the Threads team could reach the roots of negative feedback and update their platform. Similarly, during new feature launches, the in-app surveys may serve as real-time insights that help the team address bugs and glitches as soon as possible.

The model can be adapted either internally or externally. Meta's Threads Business Intelligence team (or that of any other up-and-coming social media apps) should constantly and proactively build and update the sentiment model to draw ideas on how to boost their user activeness and awareness. Additionally, Marketing Advisory firms and Customer Relationship Management (CRM) providers can expand their package offerings by adding the model as a new product insight feature.

6.2 Technical and Operational Feasibility

In general, with social media platform providers, the cost of machine learning integrations does not pose an issue. These may include cloud computing costs, CPU generators, and system adaptation processes. For smaller application companies, legacy system integration may entail initial fixed costs, but this cost can be eventually diluted as such companies expand their search and recommendation AI algorithms.

In the expansion of the model, some latency issues may appear. When Meta wishes to understand the general sentiments of users, batch processing may be used not only to reduce the complexity and operational cost of the model, but also help the business structure the insights on a periodical basis. However, when they wish to gauge the crowd response quickly following a significant event (feature launching, stock price surge…), real-time processing might be adapted.

6.3 Business Constraints and Expansion Potential

The current BERT model processes data directly on original datasets and classifies the literal meanings of reviews. However, in reality, product review datasets are often mixed with sarcasm, which may interfere with the validity of results of the model, as positive and negative meanings are reversed when taken literally. For social media applications, this problem is particularly rampant, as sarcasm can cause up to a 50% drop in prediction accuracy (Sykora et al., 2020). Therefore, potential improvements for the model can include an extra layer of preprocessing to filter genuinity. Subjectivity classification and irony detection models have been rapidly developed using a mix of Multi-task and Single-task deep learning (Pota et al., 2020), which could be integrated in the future.

Another limitation of pre-trained models for sentiment analysis tasks is the gap in quality for different languages. While the team's model is built only in English, the team recognizes that the model should ideally fit for global application. However, results and accuracy depends heavily on the language to be treated; for example, a complex and context-rich yet low-resource language like Vietnamese may suffer from more standard errors (Catelli et al., 2022). In order to address such discrepancies, state-of-the-art multilingual models that incorporate an individualized pre-processing step that normalizes noise sources from low-resource languages have been adapted to ensure better model accuracy (Pota et al., 2021). Another way to tackle the context-richness of low-resource languages is introducing a multi-label scenario in which the detection of different emotional states (anger, happiness…) may weigh differently on overall sentiment attribution, which will require more intensive fine-tuning.

REFERENCES

Amin, M.Z., & Nadeem, N. (2018). *Convolutional Neural Network: Text Classification Model for Open Domain Question Answering System*. https://doi.org/10.48550/arXiv.1809.02479

Catelli, R., Pelosi, S., & Esposito, M. (2022). Lexicon-Based vs. Bert-Based Sentiment Analysis: A Comparative Study in Italian. *Electronics*, *11*(3), 374. https://doi.org/10.3390/electronics11030374

Liao, S., Wang, J., Yu, R., Sato, K., & Cheng, Z. (2017). CNN for situations understanding based on sentiment analysis of twitter data. *Procedia Computer Science, 111*, 376–381. https://doi.org/10.1016/j.procs.2017.06.037

Pota, M., Ventura, M., Catelli, R., & Esposito, M. (2020). An effective BERT-Based pipeline for Twitter Sentiment Analysis: A case study in Italian. *Sensors*, *21*(1), 133. https://doi.org/10.3390/s21010133

Pota, M., Ventura, M., Fujita, H., & Esposito, M. (2021). Multilingual evaluation of pre-processing for BERT-based sentiment analysis of tweets. *Expert Systems With Applications*, *181*, 115119. https://doi.org/10.1016/j.eswa.2021.115119

Ray, S. (2023, July 10). Threads now fastest-growing app in history—with 100 million users in just five days. *Forbes*. https://www.forbes.com/sites/siladityaray/2023/07/10/with-100-million-users-in-five-days-threads-is-the-fastest-growing-app-in-history/?sh=4c806ba849ab

Rivera, G. (2023, August 4). *Threads' user base has plummeted more than 80%. Meta's app ended July with just 8 million daily active users*. Business Insider. https://www.businessinsider.com/threads-meta-app-decrease-daily-active-users-mark-zuckerberg-2023-8

Siddiqui, F., & Gregg, A. (2022, April 14). Elon Musk attempts hostile takeover of Twitter, calling path 'painful.' *Washington Post*. https://www.washingtonpost.com/technology/2022/04/14/elon-musk-twitter-takeover-bid/

Siddiqui, F., Lerman, R., & Merrill, J. B. (2023, April 15). A year ago, Musk asked, 'Is Twitter dying?' He may have his answer. *Washington Post.* https://www.washingtonpost.com/technology/2023/04/15/twitter-musk-bid-anniversary/

Soni, S., Chouhan, S. S., & Rathore, S. S. (2022). TextConvoNet: a convolutional neural network based architecture for text classification. *Applied Intelligence, 53*(11), 14249–14268. https://doi.org/10.1007/s10489-022-04221-9

Shuv. (2023). *Thread app dataset: 37000 entities*. Kaggle. https://www.kaggle.com/datasets/shuvammandal121/37000-reviews-of-thread-app-dataset/data

Sykora, M., Elayan, S., & Jackson, T. W. (2020). A qualitative analysis of sarcasm, irony, and related #hashtags on Twitter. *Big Data & Society*, *7*(2), 205395172097273. https://doi.org/10.1177/2053951720972735