Classify tweets with positive, neutral, or negative labels using BERT model

AML Challenge 3
Group n° 50

Victor MAYAUD[1], Elliot BOUCHY[1]

[1] Data Science, EURECOM, France

Abstract

This study explores the enhancement of sentiment analysis of text data through advanced preprocessing techniques and the use of multiple sentiment analysis models. Traditional methods of text cleaning and sentiment analysis often struggle to accurately capture the nuanced sentiments expressed in textual data, especially from social media. To address this, we employ a comprehensive preprocessing pipeline that includes various techniques for refining the text.
We evaluate the performance of different models, including transformer-based models and lexicon-based approaches, to determine the most effective method for sentiment classification. Our findings suggest that advanced models can significantly improve the accuracy of sentiment analysis compared to traditional methods. This approach promises to enhance the precision of sentiment detection in various applications, from social media monitoring to customer feedback analysis.[3]

Index Terms: Sentiment Analysis, Text Preprocessing, Natural Language Processing (NLP), BERT, Transformer Models, Machine Learning, Lemmatization, Stopwords Removal, Emoji Conversion, Tokenization, Sequence Length, Model Evaluation, Social Media Analysis

1 Introduction

In the digital era, social media platforms like Twitter have become the epicenter of public discourse, offering insights into the collective sentiments of its vast user base. Sentiment analysis, the computational study of people's opinions, emotions, and attitudes expressed through text, plays a crucial role in harnessing this voluminous data to shape business strategies, political campaigns, and public relations.

This project leverages a dataset from Figure Eight's Data for Everyone library, specifically curated to facilitate the analysis of sentiments expressed in tweets. The primary challenge is to develop a model capable of classifying tweets into three sentiment categories: positive, neutral, and negative. This classification not only aids in quantifying the emotional tone of discussions but also assists in understanding public opinion on various topics.

The dataset comprises a training set and a test set, where the training set includes labeled tweets, and the test set consists of unlabeled tweets, mimicking a real-world scenario where models must predict sentiments of unseen data. The success of the models will be evaluated using the Macro F1-Score, which considers both precision and recall, providing a balanced metric for performance across the sentiment classes.

This project not only enhances our understanding of natural language processing techniques in sentiment analysis but also explores the practical application of these techniques in a real-world social media context.

2 Dataset Analysis

2.1 Description

The dataset comprises two sets: a training set and a test set. Each set contains the columns textID, text, and selectedtext. In the training set, an additional crucial column, sentiment, categorizes each tweet as positive, negative, or neutral. This sentiment classification is fundamental for training our model to recognize and predict the emotional tone of unseen tweets. The test set lacks the sentiment column, as it is intended for model evaluation using unseen data.

One notable characteristic of the training dataset is its class imbalance. The distribution of sentiments is not uniform, as illustrated by the Class Imbalance Diagram below. There are 10,018 neutral tweets, compared to 7,003 negative tweets and 8,711 positive tweets. This imbalance can influence the model's performance, potentially leading it to be biased toward the more frequently occurring classes. Addressing this imbalance will be critical in developing a robust model that performs well across all sentiment classes.

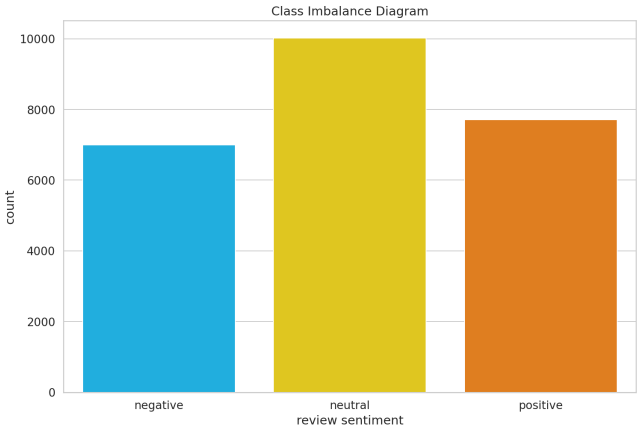


Figure 1. Distribution of Sentiment Classes in the Training Dataset

2.2 Analysis

The sentiment analysis dataset is composed of concise messages similar to those typically found on social networks for comments or publications, such as on Twitter. These messages vary significantly in length and complexity, as indicated by the analysis of the dataset. The maximum length of tweets in characters is 141, while the maximum word count reaches 38. On average, tweets are composed of approximately 68 characters and

14 words. This variability in message length reflects the diverse nature of expression on social platforms, ranging from short declarative statements to more elaborated expressions.

Further analysis reveals distinct patterns in message length distribution across different sentiments. The 'Count distribution by length in Tweets' graph demonstrates how negative and positive sentiments tend to have longer messages on average, possibly due to the expressive nature of such sentiments. Neutral sentiments, while still varied, tend to feature shorter and more concise messages. These patterns are crucial for modeling, as they suggest that message length may be an indicative feature of sentiment.

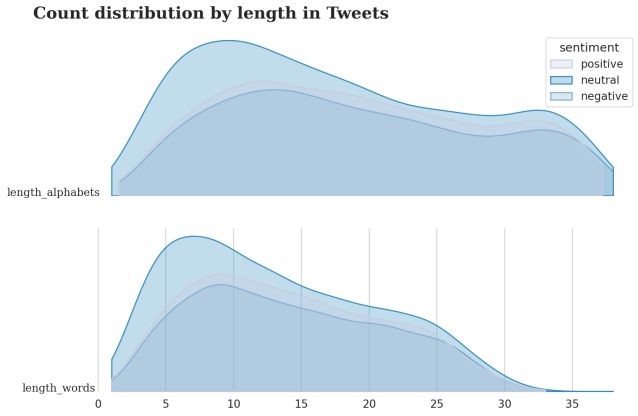


Figure 2. Comparative Distribution of Tweet Lengths by Sentiment Category

The WordCloud by sentiment per selected text in Tweets' visually represents the most frequent words used in different sentiments. Negative texts frequently include words like 'sick', 'sorry', and 'bad', indicating a clear linguistic pattern of dissatisfaction or discomfort. Neutral texts show a prevalence of general and non-emotive terms like 'today' and 'going', reflecting their impartial nature. Positive texts, on the other hand, often contain words such as 'love', 'good', and 'happy', highlighting the positive affirmations commonly expressed under this sentiment.

Figure 3. Wordcloud by sentiment per selected text in tweets

These insights are pivotal for developing a nuanced sentiment analysis model that can accurately classify and understand the underlying tones in social media text data.

2.3 Pre-processing

Data processing is a crucial step in improving the accuracy of sentiment analysis using natural language processing models. To this end, we have applied a series of pre-processing techniques to clean and normalize the textual data before using it to train the BERT model. The specific steps are as follows:

- **Stopwords Removal:** We removed commonly used empty words (such as "a", "an", "the") as they do not bring significant value to sentiment analysis.
- **Lemmatization with Spacy:** We used the Spacy library to perform lemmatization, which consists of transforming each word into its basic form or lemma. This normalizes grammatical variations in words.
- **Removing URLs:** URLs present in tweets have been removed as they do not contribute to sentiment analysis and can introduce noise.
- **Conversion of emojis and emoticons to words:** Emojis and emoticons have been converted to their text equivalents to better capture the sentiments they express.
- **Spell Checker:** A spell checker was used to correct common spelling errors in tweets, improving the quality of textual data.
- **Handling words with multiple repetition letters:** Words containing excessively repeated letters have been normalized (for example, "cooool" becomes "cool").
- **Chat Words Conversion:** Abbreviations and informal language commonly used in chats and tweets have been converted to their full form (for example, "u" becomes "you").
- **Twitter handlers deletion:** Twitter handles (@username) have been removed from tweets to avoid user-specific bias.
- **Numeric data removed:** Numbers and numeric data have been removed as they are not relevant to sentiment analysis in this context.
- **Removal of multiple white spaces and spaces at the beginning and end of text:** We have eliminated superfluous white spaces and spaces at the beginning and end of text to standardize the data.
- **Removal of punctuation after emoticon processing:** Once the emoticons had been converted to words, the remaining punctuation was removed to avoid confusion in word analysis.

These pre-processing steps clean and normalize the text data, making it more consistent and better suited to training the BERT model for accurate sentiment analysis.

3 First Proposed Approach

3.1 TextBlob

TextBlob [4] is a popular Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. In the context of sentiment analysis, TextBlob uses a lexicon-based approach, where each word in the text is associated with sentiment scores. The overall sentiment score for a text is derived by aggregating the scores of individual words. This makes it particularly useful for straightforward sentiment predictions on relatively clean and formal datasets, offering a quick and intuitive method for assessing sentiments.

3.2 VADER

VADER (Valence Aware Dictionary and sEntiment Reasoner) [5] is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of a sentiment lexicon, which is a list of lexical features (e.g., words), annotated with their sentiment strength, and a set of five heuristic rules to handle different contexts and grammatical nuances. These rules account for factors

such as punctuation, capitalization, degree modifiers, and conjunctions, enhancing its ability to understand the sentiment in complex and informal text. Due to its design, VADER is exceptionally good at handling social media text, as well as texts from other domains that are casual or highly expressive.

4 Results

4.1 Model Performance

The performance of the TextBlob and VADER models in sentiment analysis can be quantitatively assessed through their Macro F1-Scores. Below is a summary of their performance metrics:

Model	Macro F1-Score
TextBlob	0.60
VADER	0.64

Table 1
Macro F1-Scores of Sentiment Analysis Models

The Macro F1-Scores indicate that both models perform similarly, with VADER slightly outperforming TextBlob. This relatively close performance suggests that both models are capable of effectively categorizing the sentiments of text data, albeit with some limitations in accuracy and consistency.

To further analyze model effectiveness, the confusion matrices for both TextBlob 4 and VADER 5 reveal interesting insights into their prediction capabilities. The confusion matrix for VADER shows a more balanced distribution of predictions across the three sentiment classes, albeit with some challenges in correctly identifying neutral sentiments. In contrast, TextBlob tends to misclassify negative sentiments as neutral more frequently, which could be indicative of its sensitivity to less pronounced sentiment expressions.

These matrices will be detailed further in the following sections, providing a visual representation of where each model succeeds and where it tends to make errors. This visualization aids in understanding the nuanced performance of each model, especially in how they handle boundary cases between sentiment categories.

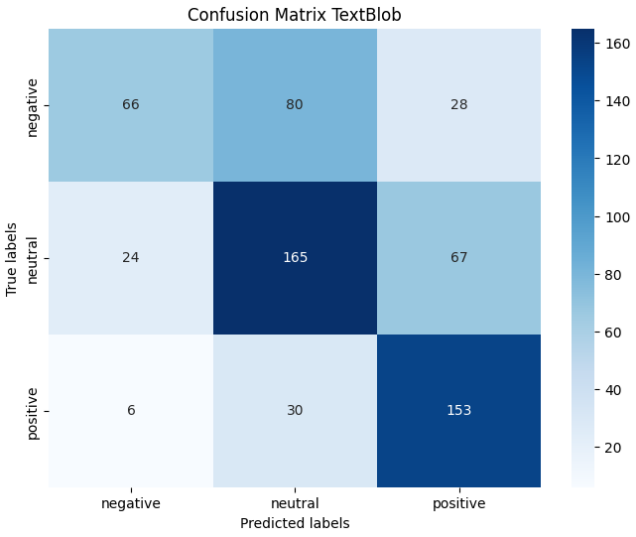


Figure 4. Confusion Matrix Textblob

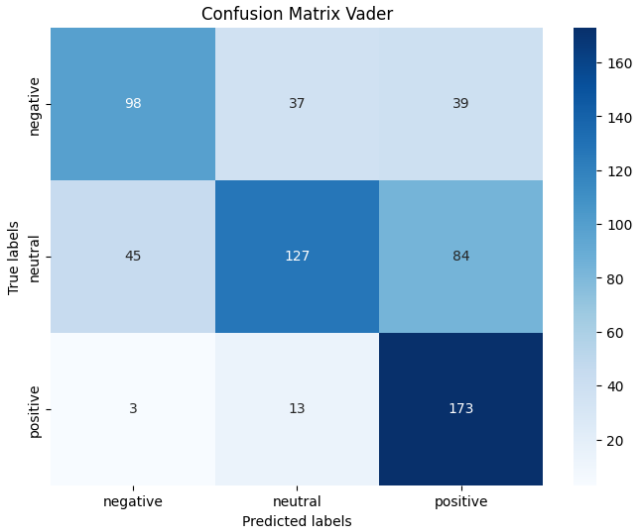


Figure 5. Confusion Matrix Vader

4.2 Discussion of Difficulties

Both models have shown reasonable effectiveness in classifying sentiments into positive, neutral, and negative categories. However, there are inherent difficulties associated with the accuracy of these classifications, particularly around the threshold used to determine neutrality. In both models, any sentiment score between -0.1 and 0.1 is considered neutral. This threshold can be too restrictive and potentially misclassify subtle expressions of sentiment.

For example, very mild positive or negative sentiments often expressed with slight positivity or negativity can be categorized as neutral due to the narrow range set for neutrality. Such misclassifications can be observed in the confusion matrices for both models, where a significant number of neutral predictions should have been positive or negative. This is a limitation in the current sentiment scoring system that could be mitigated by adjusting the thresholds or by enhancing the contextual understanding of the models.

5 Second Proposed Approach

5.1 Model Architectures

After cleaning and pre-processing the text data, we used the BERT (Bidirectional Encoder Representations from Transformers) model for sentiment analysis. BERT is known for its ability to understand the context of words in a sentence, thanks to its bidirectional transformer architecture.[1]

We've explored several versions of pre-trained BERT models available via the Hugging Face library. These included cased and uncased versions. After several tests, we found that the cased version of BERT provided better results for sentiment analysis, due to the importance of case in conveying emotion (e.g. "BAD" vs. "bad").

BERT is built on the Transformer architecture, which consists of an encoder and a decoder. However, BERT uses only the encoder part of the Transformer. The encoder processes the input text through multiple layers of self-attention and feed-forward neural networks. Each layer refines the representation of the input text by considering the relationships between all words in a sentence.

BERT operates in two phases: pre-training and fine-tuning.

- **Masked Language Model (MLM):** Randomly masks some of the tokens in the input, and the model is trained to predict these masked tokens based on their context.
- **Next Sentence Prediction (NSP):** The model is trained to predict whether a given pair of sentences follows each other in the text.

This pre-training allows BERT to learn a deep understanding of language structure and context.

In the context of sentiment analysis, BERT is fine-tuned on a labeled dataset of text (e.g., tweets) where each text is associated with a sentiment label (positive, negative

, neutral, etc.). The steps involved in applying BERT to sentiment analysis are as follows:

- **Tokenization:** The pre-processed data were converted into token sequences using the BERT tokenizer. Each tweet was tokenized into a sequence of IDs corresponding to the words and subwords of the BERT model. To ensure consistent input to the model, we set a maximum sequence length. Shorter sequences were padded, while longer sequences were truncated.

5.2 Model training

- **Splitting the Data:** The data were divided into training, validation and test sets ((23495) (619) (618)). The splitting was done in such a way as to maintain a balanced distribution of sentiment classes in each set.
- **Hyperparameter configuration:** We configured the hyperparameters essential for training the model, including learning rate, batch size and number of training epochs. An optimal learning rate was chosen to balance learning speed and convergence stability.
- **Training:** The BERT model was trained using a loss function adapted to sentiment classification. We used the Adam optimizer with learning rate decay to refine the model weights. Regularization techniques, such as dropout, were applied to avoid overfitting.

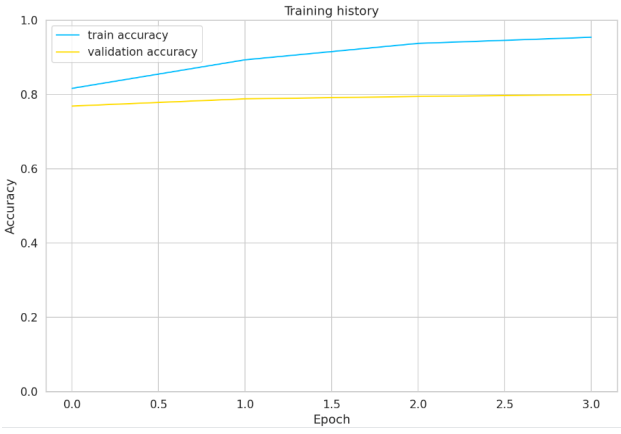


Figure 6. Training history

Initially, we trained the model for 5 epochs. However, by analyzing the model's performance after each epoch, we noticed that the validation accuracy reached a plateau from the third epoch onwards. Specifically, we observed that validation accuracy no longer increased significantly after three epochs. This observation suggests that the model reaches its optimal learning capacity after three epochs, and continuing training beyond this point brings no further improvement in terms of validation accuracy. Consequently, to avoid overfitting and optimize the use of computational resources, we decided to limit training to three epochs.

We measured the model's performance at each epoch using Jaccard's coefficient[6], a metric commonly used to assess similarity between two sets. The results obtained on the training and validation datasets are summarized below.

Table 2
Jaccard coefficients as a function of the number of epochs

Epoch	Train Jaccard	Validation Jaccard
1	0.8466	0.7712
2	0.8802	0.7526
3	0.9306	0.7826

In the first epoch, the model performs relatively well on both datasets, with close Jaccard coefficients for training and validation. However, we observe a decrease in the Jaccard coefficient on the validation set in the second epoch, although the training Jaccard coefficient increases.

This decrease can be attributed to several factors:

- **Early overfiting:** The model may start to overlearn the specific features of the training dataset, losing generalizability. This leads to a drop in performance on the validation set.
- **Data variability:** The distribution of data in the validation set may differ slightly from that in the training set. As the model adjusts more and more to the training data, it may perform less well on unseen data.
- **Hyperparameter adjustment:** It is possible that some hyperparameters of the model or training process,leading to fluctuations in validation performance.

The model shows continuous improvement on the training set over the epochs, which is expected. However, the decline in validation performance in epoch two highlights the importance of monitoring validation metrics to avoid overlearning. In epoch three, validation performance improves again, suggesting that the model is beginning to stabilize and generalize better.

5.3 Model evaluation

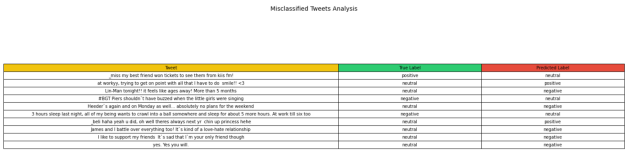
After training, we evaluated the model's performance on the test set. Evaluation metrics included accuracy, precision, recall and F1 score, providing an overview of the model's ability to correctly classify sentiments in tweets.

Table 3
Performance Metrics for Sentiment Analysis Using BERT

Class	Precision	Recall	F1-Score	Support
Negative	0.77	0.80	0.79	174
Neutral	0.74	0.75	0.74	256
Positive	0.83	0.78	0.81	189
Accuracy		0.77		619
Macro Avg	0.78	0.78	0.78	619
Weighted Avg	0.78	0.77	0.77	619

We also analyzed tweets where the model made errors, to understand the limitations and challenges of sentiment analysis with BERT. This analysis helps to identify where further improvements are needed.

- **Ambiguity in language:** Sarcastic or ironic tweets are particularly difficult for the model to detect without deep contextual understanding.



Tweet	True Label	Predicted Label
yours my best friend won tickets to see them from kiss fm!	positive	neutral
at workyy, trying to get on point with all that I have to do :smile!: <3	neutral	positive
Lin-Man tonight!? it feels like ages away! More than 5 months	neutral	negative
IBIG!! Piers shouldn't have buzzed when the little girls were singing	negative	neutral
Heidiiii' s again and on Monday as well, , absolutely no plans for the weekend	neutral	negative
3 hours sleep last night, all of my being wants to crawl into a ball somewhere and sleep for about 5 more hours. At work till six too	negative	neutral
Jeni Haha yeah u did, oh well theres always next yr :chin up princess hehe	neutral	positive
James and I battle over everything too! It' s kind of a love-hate relationship	neutral	negative
I like to support my friends It' s sad that I' m your only friend though	neutral	negative
yes. Yes you will.	neutral	negative

Figure 7. missclassified tweet analysis

- **Linguistic variability:** The presence of dialects, jargon or typos can make prediction more difficult.
- **Missing context:** Some tweets require an external context for correct interpretation, which may not be taken into account by the model.

In conclusion, poorly predicted tweets are often the result of the complexity and linguistic diversity of Twitter data. Specific challenges include the management of irony, dialect variations and informal language. In addition, the choices made during data cleaning and pre-processing can also have a significant impact on model performance. A better understanding and more sophisticated handling of these aspects can potentially improve prediction accuracy.

Interpreting the confusion matrix allows us to identify the types of error that the model makes most frequently. The confusion matrix below illustrates the performance of our BERT model in classifying tweets into three categories: negative, neutral and positive.

The model performs well overall, with a high number of true positive and negative predictions. The most frequent errors are between neutral and positive classes, where 37 positive tweets were classified as neutral and 27 neutral tweets were classified as positive. Errors between negative and neutral classes were also significant, with 31 negative tweets classified as neutral and 38 neutral tweets classified as negative.

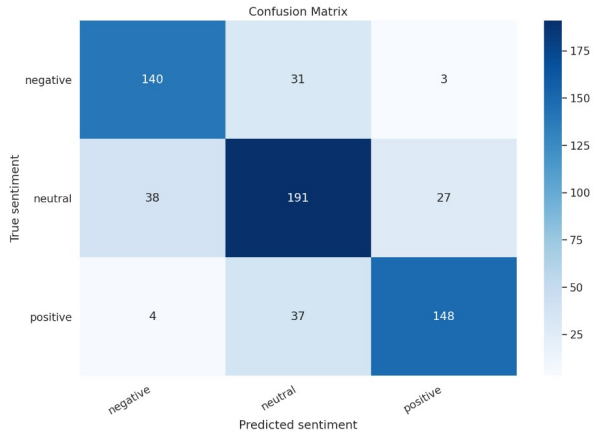


Figure 8. missclassified tweet analysis

6 Comparative Analysis of Sentiment Analysis Models

The sentiment analysis models employed in this study—TextBlob, VADER, and BERT—exhibit varying levels of performance across different metrics. The F1-Score, which combines precision and recall into a single metric, is particularly useful for comparing these models given the class imbalances in our dataset. The following table summarizes the Macro F1-Scores for each model:

Model	Macro F1-Score
TextBlob	0.60
VADER	0.64
BERT	0.78

Table 4
Comparison of Macro F1-Scores Across Models

BERT demonstrates a superior performance with a Macro F1-Score of 0.78, outpacing both TextBlob and VADER. This enhancement can be attributed to BERT's deep learning framework, which better captures the context within the text through its transformer architecture. In contrast, TextBlob and VADER, while effective for simpler applications, struggle with more complex sentiment expressions due to their primarily lexicon-based approaches.

VADER, designed specifically for social media texts, shows better results than TextBlob, reflecting its ability to interpret the nuances of informal language, including slang and emoticons. However, it still falls short of BERT's deeper contextual understanding, which allows for a more accurate classification of sentiments, particularly in nuanced or mixed expressions.

This comparison underscores the importance of choosing an appropriate model based on the specific characteristics and requirements of the sentiment analysis task, especially considering the complexity and variety of the text involved.

7 Discussion

Overall, BERT outperformed both VADER and TextBlob in terms of precision, recall, and F1-score. This superior performance can be attributed to BERT's deep learning architecture, which is capable of understanding the context and subtleties in the text better than rule-based models. However, the increased complexity and resource requirements of BERT must be considered when choosing a model for practical applications.

while all three models - BERT, VADER, and TextBlob - showed effectiveness in sentiment analysis, BERT was the most accurate and reliable. The choice of model should consider the specific requirements and constraints of the application, balancing the need for accuracy with computational resources. For applications where accuracy is paramount and resources are available, BERT is the preferred choice. However, for real-time applications or environments with limited resources, VADER and TextBlob offer viable alternatives.

7.1 Limitations of the Approach

While our approach using BERT, VADER, and TextBlob for sentiment analysis has demonstrated effectiveness, several limitations need to be acknowledged:

- **Computational Complexity** The BERT model, despite its superior performance, is computationally expensive. It requires significant hardware resources and time for training and inference. This limitation makes it less feasible for real-time applications or scenarios with limited computational resources.
- **Data Dependency** BERT's performance is highly dependent on the quality and quantity of the training data. Insufficient or biased training data can lead to suboptimal performance and generalization issues.
- **Rule-Based Limitations** Both VADER and TextBlob rely on predefined rules and lexicons. This reliance limits their ability to capture context and nuanced sentiments, especially in complex or non-standard text.

- **Language and Domain Specificity** The models may not perform equally well across different languages or domains without additional customization and retraining. This specificity can limit the generalizability of the models to new or varied contexts.
- **Sentiment Complexity** Sentiment analysis often reduces the rich complexity of human emotions to simple categories (positive, neutral, negative). This reduction can overlook subtleties and mixed emotions present in the text, leading to less accurate classifications.

7.2 Future work

To address the limitations and further enhance the effectiveness of our sentiment analysis approach, several avenues for future work are proposed. We could exploring more efficient versions of BERT, such as DistilBERT or BERT with pruning techniques, can help reduce computational requirements while maintaining high performance, extending the models to support multiple languages through multilingual BERT models or training on multilingual datasets can enhance their applicability in global contexts.Incorporating models that better capture the context, such as transformer models with improved context handling or hybrid approaches combining rule-based and machine learning methods, can enhance sentiment detection accuracy. Or moving beyond simple sentiment analysis to detect a wider range of emotions (e.g., joy, anger, sadness) can provide a more nuanced understanding of the text.

8 Conclusion

This project showcases the efficacy of advanced machine learning models, especially BERT, in sentiment analysis of social media content. The study employed various sentiment analysis models including TextBlob, VADER, and BERT, to classify tweets into positive, neutral, and negative sentiments.

The results confirmed that BERT outperformed other models in terms of precision, recall, and F1-score, demonstrating its superior capability to understand the context and subtleties of textual data. This is attributed to its deep learning architecture and the utilization of transformer mechanisms which provide a more nuanced understanding of language.

Despite its advantages, the research also highlighted several limitations associated with the models, particularly BERT. Its high computational requirements and dependency on large, well-curated datasets pose challenges for real-time applications and environments with limited resources. The rule-based models like TextBlob and VADER, while less resource-intensive, struggle with complex language expressions and the nuanced sentiment found in informal social media text.

Future work suggested includes exploring more efficient versions of BERT, such as DistilBERT, to reduce computational demands. Additionally, extending the models to support multiple languages and incorporating more advanced context-capturing capabilities could enhance their applicability and accuracy in diverse global contexts.

Acknowledgements

This research received support during the AML course, instructed by Professor Pietro MICHIARDI, Head of the Data Science Department at EURECOM, France.

References

[1] BERT 101 https://huggingface.co/blog/bert-101

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.

[3] What Is sentiment analysis ? https://www.freecodecamp.org/news/what-is-sentiment-analysis-a-complete-guide-to-for-beginners/

[4] Textblob https://textblob.readthedocs.io/en/dev/

[5] Vader https://www.researchgate.net/publication/275828927_VADER_A_Parsimonious_Rule-based_Model_for_Sentiment_Analysis_of_Social_Media_Text

[6] Exploring Jaccard Similarity and Distance with Python https://www.adventuresinmachinelearning.com/exploring-jaccard-similarity-and-distance-with-python/