Twitter Data Pre-processing and Sentiment Analysis Project

Daniil Gurgurov[*] **Katharina Trinley**[†]
Saarland University

Abstract

This report is focused on reviewing the Twitter data pre-processing and sentiment analysis done for the Tools4NLP final project. Initially, we discuss our pre-processing, issues found within the data, and what tools we used and why. Thereafter, we look into the difference between VADER and TextBlob. In addition, we look at the Twitter RoBERTa sentiment analysis model for a bonus point. Finally, we apply the aforementioned sentiment analysis tools to our data.

1 Introduction

In this report, we dive into a comprehensive review of the Twitter data preprocessing performed for the Tools4NLP assignments. Our focus is on specifying the steps involved in preprocessing, addressing encountered issues within the data, and providing insights into the choice of tools and methodologies applied. Furthermore, a comparative analysis of sentiment analysis tools is presented, highlighting the distinctions between VADER (Hutto and Gilbert, 2014) and TextBlob (Loria, 2018). The ultimate aim is to apply these models to our preprocessed dataset to understand how well they work on the modern tweets. In addition, we perform sentiment analysis of the data using one of the models available on HuggingFace.

2 Dataset

The dataset, extracted from Twitter and organized into a DataFrame, captures sentiments expressed in tweets related to different topics. The DataFrame consists of several columns, including 'number', 'account,' 'sentiment', and 'tweet'. The 'number' column serves as a unique identifier, while 'account' specifies the associated Twitter account. The 'sentiment' column categorizes tweets into classes such as 'Positive,' 'Negative,' 'Neutral,' and 'Irrelevant,' reflecting the expressed sentiments. The 'tweet' column contains the actual text content of the tweets.

Upon inspection of the summary statistics, the dataset comprises 74,682 entries, with a mean 'number' value of approximately 6433. The 'number' values range from 1 to 13,200, and the standard deviation is 3740.43. Sentiment distribution shows a predominance of 'Negative' and 'Positive' sentiments, with 22,542 and 20,832 instances, respectively. 'Neutral' sentiments are not much less, accounting for 18,318 tweets, while 'Irrelevant' sentiments constitute 12,990 entries. The overall sentiment distribution before the data preprocessing can be observed in Figure 1. These descriptive statistics offer a comprehensive overview of the sentiment distribution within the dataset, providing us with valuable insights for the following sentiment analysis tasks.

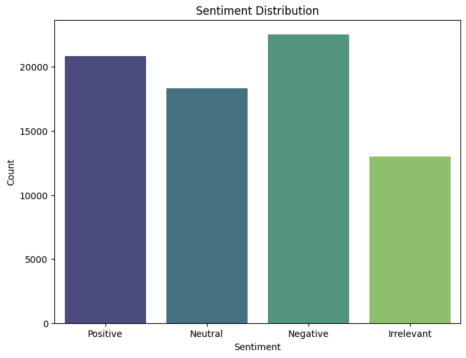

Figure 1: Sentiment Distribution Before Data Preprocessing

3 Data Preprocessing

In this section, we describe the data preprocessing steps performed on the dataset to prepare it for sentiment analysis. The dataset is loaded from a CSV file, and various preprocessing techniques are

[*]dagu00001@stud.uni-saarland.de
[†]katr00001@stud.uni-saarland.de

applied using Python. The following steps were carried out:

3.1 Handling Missing Values

Firstly, any missing values in the 'tweet' column were addressed by dropping the corresponding rows. The number and percentage of missing values in column column are 686 and 0.92% respectively. Removing these columns was the easiest way of dealing the missing values and does not result in a big loss of data.

3.2 Converting Data Types

The data types of the columns in the DataFrame were then converted for consistency. The 'number' column was converted to integer type, and the 'account', 'sentiment', and 'tweet' columns were converted to string type.

3.3 Lowercasing Text

The 'tweet', 'account', and 'sentiment' columns were lowercased to ensure uniformity and ease of analysis. This step is crucial for sentiment analysis as it reduces the impact of case variations in text.

3.4 Removing Non-ASCII Characters and Emojis

Non-ASCII characters and emojis were removed from the 'tweet' column. This helps in cleaning the text data and excluding potential noise that might affect sentiment analysis results.

3.5 Removing Stopwords

Stopwords, common words that do not contribute much to the sentiment, were removed from the 'tweet' column as well. This step is beneficial for focusing on more meaningful words and improving the accuracy of sentiment analysis.

3.6 Stemming Words

Words in the 'tweet' column were stemmed using the Porter Stemmer algorithm. This process involves reducing words to their root form, aiding in capturing the essence of the text without the complexity of different word forms.

3.7 Removing Numbers and Punctuation

Numbers and punctuation were removed from the 'tweet' column to simplify the text for sentiment analysis. This step ensures that numerical values and punctuation marks do not interfere with the sentiment analysis process.

3.8 Removing Non-English Words

Non-English words were removed from the 'tweet' column to focus on sentiments expressed in the English language. This step contributes to the overall accuracy of sentiment analysis.

3.9 Fixing Labels

The 'sentiment' labels were fixed by removing the 'irrelevant' label entries as they do not fall into any of the other categories. This step simplifies the sentiment classes distribution for analysis.

3.10 Removing Empty Tweets

Rows with empty tweets were removed from the dataset, ensuring that only meaningful data is considered in sentiment analysis.

These preprocessing steps collectively contribute to creating a clean and standardized dataset, enhancing the effectiveness of sentiment analysis models.

4 Sentiment Analysis Tools

Further, we look at different sentiment analysis models utilized for the project.

4.1 NLTK (VADER)

NLTK (Bird et al., 2009), a comprehensive library for natural language processing, includes modules for sentiment analysis. It provides various tools and resources for text processing, making it a versatile choice for researchers and developers. One of NLTK's sentiment analysis tools is VADER. VADER is a lexicon and rule-based sentiment analysis tool designed specifically for social media text. Developed within NLTK, VADER excels at handling informal language, emoticons, and acronyms commonly found in online communication. Its strength is in recognizing nuanced sentiments, including sarcasm and irony. VADER assigns polarity scores to text, indicating the strength and direction of sentiment. It is widely adopted for its out-of-the-box functionality and especially useful for quick sentiment assessments in social media monitoring and customer feedback analysis.

4.2 SpaCy (TextBlob)

SpaCy (Honnibal and Montani, 2017) is another popular natural language processing library that has gained recognition for its efficiency and speed in various language processing tasks, including sentiment analysis. While SpaCy itself doesn't provide

a dedicated sentiment analysis module, it has a support for a TextBlob module, designed for this purpose. TextBlob is a simple and user-friendly lexicon-based tool with some predefined rules for sentiment analysis. It provides a high-level interface for various natural language processing tasks, including sentiment polarity analysis. TextBlob's sentiment module returns polarity and subjectivity scores, which makes it easy for users to interpret sentiment and emotional strength. This tool is particularly advantageous for those who wish for a quick implementation of sentiment analysis without diving into technical details. While TextBlob lacks the depth of customization offered by more complex models, its simplicity and ease of use make it a good choice for quick sentiment assessments.

4.3 Twitter-roBERTa-base for Sentiment Analysis (Bonus)

This subsection introduces the Twitter-roBERTa-base model (Loureiro et al., 2022) for sentiment analysis available on HuggingFace. The model is based on RoBERTa-base Large Language Model architecture and was trained on approximately 124 million tweets from January 2018 to December 2021. It is a Transformer(Vaswani et al., 2017) based model with an Encoder-only architecture. It has been fine-tuned for sentiment analysis using the TweetEval benchmark.

The original Twitter-based RoBERTa model can be accessed on Hugging Face's model hub: [1].

The sentiment labels used by this model are:

- 0: Negative
- 1: Neutral
- 2: Positive

5 Sentiment Analysis Results

In this section, we present the results of sentiment analysis using three different models: VADER, SpacyTextBlob, and Hugging Face. The dataset used for evaluation contains a total of 59,863 samples, with sentiments labeled as negative, neutral, and positive. It has been preprocessed using the steps described in the previous parts.

[1] https://huggingface.co/cardiffnlp/twitter-roberta-base-2021-124m

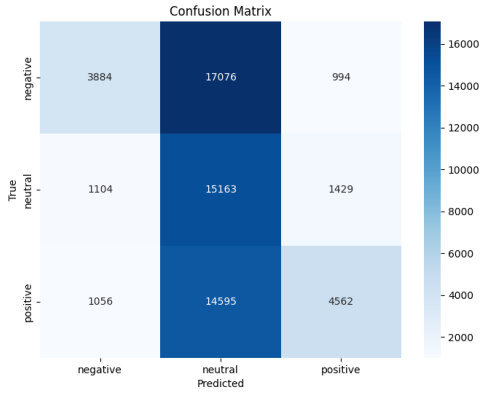

Figure 2: Confusion Matrix - VADER Sentiment Analysis

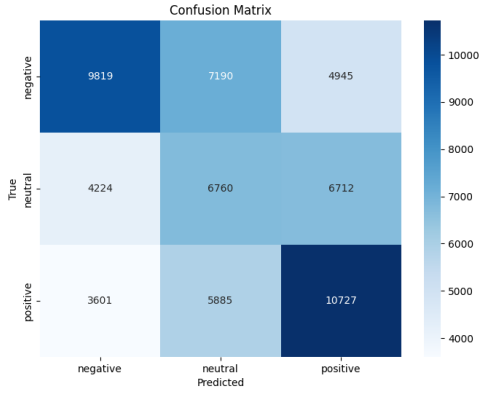

Figure 3: Confusion Matrix - SpacyTextBlob Sentiment Analysis

5.1 VADER Sentiment Analysis

The classification report for VADER Sentiment Analysis is shown in Table 1.

5.1.1 Analysis of VADER Results

VADER achieves an overall average F1-score of 0.36. This score shows the model's ability to balance precision and recall across sentiment classes. However, it is important to note that the model struggles to accurately predict negative sentiments, as indicated by the low recall for the negative class.

Sentiment	Precision	Recall	F1-Score
Negative	0.64	0.18	0.28
Neutral	0.32	0.86	0.47
Positive	0.65	0.23	0.34
Average			**0.36**

Table 1: VADER Sentiment Analysis Results

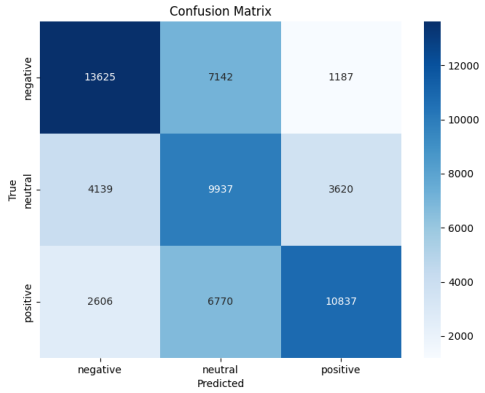

Figure 4: Confusion Matrix - RoBERTa Sentiment Analysis

Sentiment	Precision	Recall	F1-Score
Negative	0.56	0.45	0.50
Neutral	0.34	0.38	0.36
Positive	0.48	0.53	0.50
Average			**0.45**

Table 2: SpacyTextBlob Sentiment Analysis Results

5.2 SpacyTextBlob Sentiment Analysis

The classification report for SpacyTextBlob Sentiment Analysis is presented in Table 2.

5.2.1 Analysis of SpacyTextBlob Results

SpacyTextBlob demonstrates balanced performance across sentiment classes, with an average F1-score of 0.45. The model excels in predicting negative sentiments, while improvements could be made in precision for the neutral class.

5.3 Hugging Face Sentiment Analysis

The classification report for RoBERTa Sentiment Analysis is presented in Table 3

5.3.1 Analysis of Hugging Face Results

Hugging Face achieves an overall average F1-score of 0.57. The model shows competitive performance across sentiment classes, with notable precision for negative and positive sentiments. The model demonstrates balanced recall for neutral sentiments.

Sentiment	Precision	Recall	F1-Score
Negative	0.67	0.62	0.64
Neutral	0.42	0.56	0.48
Positive	0.69	0.54	0.60
Average			**0.57**

Table 3: Hugging Face Sentiment Analysis Results

6 Discussion

The results of the sentiment analysis using VADER, TextBlob, and Hugging Face's RoBERTa reveal distinct performance characteristics across the models. This discussion aims to analyze the outcomes and identify factors contributing to RoBERTa's better performance.

6.1 Comparison of Sentiment Analysis Models

Starting with VADER, the model demonstrates challenges in accurately predicting negative sentiments, as indicated by the low recall for this class (0.18). This weakness might be explained by the lexicon-based nature of VADER, which may struggle with understanding context in complex language expressions.

SpacyTextBlob, on the other hand, shows more balanced performance across sentiment classes, with an F1-score of 0.45. While the model excels in predicting negative sentiments, there is room for improvement in precision for the neutral class.

Hugging Face's RoBERTa emerges as the top-performing model in this analysis, achieving an average F1-score of 0.57. The model demonstrates competitive precision and recall across all sentiment classes, showing its ability to capture the sentiment in tweets.

6.2 Factors Contributing to RoBERTa's Superior Performance

Several factors contribute to RoBERTa's superior performance over VADER and SpacyTextBlob. One key factor is RoBERTa's deep learning architecture, which enables the model to learn complex patterns and representations in the data. Unlike lexicon-based approaches, RoBERTa can capture contextual information and adapt to variations in language usage.

Additionally, RoBERTa has been pretrained on a large corpus of tweets, allowing it to utilize a rich understanding of Twitter-specific language nuances. This pretraining contributes to the model's robustness in dealing with informal language, slang, and abbreviations often found in tweets.

Furthermore, RoBERTa's fine-tuning on sentiment analysis tasks, specifically using the TweetEval benchmark, enhances its performance for sentiment classification.

7 Conclusion

This report covered an exploration of Twitter data preprocessing and sentiment analysis, focusing on the comparison of sentiment analysis models VADER, SpacyTextBlob, and Hugging Face's RoBERTa.

The data preprocessing phase involved addressing missing values, converting data types, lowercasing text, removing non-ASCII characters, emojis, stopwords, stemming words, numbers, punctuation, non-English words, fixing labels, and removing empty tweets. These steps collectively contributed to creating a clean and standardized dataset for effective sentiment analysis.

The subsequent sentiment analysis using VADER, SpacyTextBlob, and RoBERTa provided valuable insights into the performance characteristics of each model. VADER and TextBlob, being lexicon and rule-based approach models, struggled with nuanced sentiment expressions. RoBERTa, expectedly, was the top-performing model, demonstrating superior precision, recall, and overall F1-score across all sentiment classes.

The discussion further delved into the factors contributing to RoBERTa's superior performance, emphasizing its deep learning architecture, pre-training on a large and relevant dataset, and fine-tuning for sentiment analysis tasks.

In conclusion, this report emphasizes the importance of robust data pre-processing in preparing Twitter data for sentiment analysis. It underscores the limitations of traditional lexicon-based and rule-based models like VADER and SpacyTextBlob and showcases the advancements achieved with state-of-the-art deep learning models like RoBERTa.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

Steven Loria. 2018. textblob documentation. *Release 0.15*, 2.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.