AML challege: Sentiment Analysis

Giuseppe Antonio Orlando
Politecnico di Torino
EURECOM
orlando@eurecom.fr

Simone Varriale
Politecnico di Torino
EURECOM
varriale@eurecom.fr

Gabriele Sanmartino
Politecnico di Torino
EURECOM
sanmarti@eurecom.fr

I. Introduction

The task of sentiment analysis involves determining the emotional tone behind a body of text. This is crucial for various applications, including social media monitoring, customer feedback analysis, and opinion mining. The objective for this task is the development and evaluation of a sentiment classification model.

This report outlines the dataset and pre-processing methods used, describes the model selection process, presents the performance of different pre-processing techniques, and evaluates the final model. Additionally, an analysis of advanced data augmentation techniques using a large language model is presented and the attention weights of the model are analyzed, to identify the most influential words in sentiment classification.

II. Dataset

The provided dataset contains textual data with corresponding sentiment annotations. Each entry in the dataset consists of the following fields:

- textID: A unique identifier for each text entry.
- text: The full original text from which the selected text and sentiment are derived.
- selected_text: A portion of the original text that is highlighted as the most indicative of the sentiment.
- sentiment: The sentiment label associated with the selected text. The possible values are:
  - positive (0): Indicates a positive sentiment.
  - neutral (1): Indicates a neutral sentiment.
  - negative (2): Indicates a negative sentiment.

Some examples of text and sentiment are shown in Table I. The distribution of the sentiments is instead shown in Figure 1.

The dataset will be split in training, validation and test data with respective percentages: 70%, 20%, 10%.

III. Data pre-processing

Text data pre-processing is a crucial step in natural language processing (NLP) that transforms raw text into a format suitable for analysis and modeling. This section describes the techniques that will be employed:

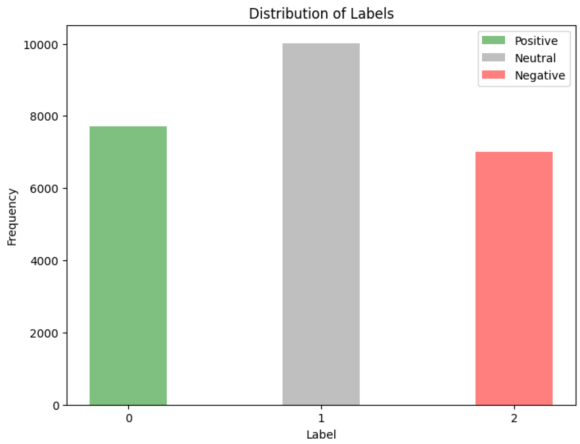


Fig. 1: Distribution of samples

stop words removal, stemming, lemmatization, and data augmentation.

A. Stop Words Removal

Stop words are commonly occurring words in a language that carry minimal semantic value, such as ”and,” ”the,” and ”is.” Removing stop words can significantly reduce the dimensionality of the text data without losing meaningful information. This simplification of the text helps the models to focus on more informative parts of the input samples and should improve the overall performances.

B. Stemming

Stemming [2] is the process of reducing words to their base or root form. This is done by removing suffixes from words to achieve a common base form. For example, the words ”running,” ”runner,” and ”runs” can be reduced to the root word ”run”. Stemming helps in normalizing the text by reducing the number of distinct terms and improving the efficiency of text processing. Among the different stemmers, Porter Stemmer has been selected for the task.

text	sentiment
good luck with your auction	positive
Hmm..You can‘t judge a book by looking at its cover	neutral
Hello, yourself. Enjoy London. Watch out for the Hackneys. They‘re mental.	negative

TABLE I: Example sentences with their sentiments.

C. Lemmatization

Lemmatization [2] is similar to stemming but involves reducing words to their dictionary or base form, known as the lemma. Unlike stemming, lemmatization considers the context and morphological analysis of words. For example, the word ”better” is lemmatized to ”good.” Lemmatization often yields more accurate and meaningful base forms compared to stemming and it helps to improve the quality and coherence of the text data. The SpaCy Lemmatizer has been selected for the task.

D. Data Augmentation

Data augmentation involves generating additional training samples from the existing data to reduce overfitting and improve model generalization. In the context of text data, augmentation techniques can include [5]:

- *Synonym Replacement*: Replacing words with their synonyms to create new sentences. For example, ”The quick brown fox jumps over the lazy dog” can be augmented to ”The fast brown fox leaps over the lazy dog.”
- *Insertion*: Adding new words or phrases into sentences. For example, adding adjectives or adverbs can create new samples.
- *Random Swap*: Randomly choosing two words in the sentence and swapping their positions.
- *Random Deletion*: Randomly removing words from the sentence to create shorter, varied versions of the text.

These techniques might help to create a more robust and diverse training dataset, which can enhance the performances and generalization abilities of NLP models. *Insertion* has been picked as the data augmentation technique for the task.

IV. Model Selection

BERT is a transformers model pre-trained on a large corpus of English data in a self-supervised fashion. This means it was pre-trained on the raw texts only, with no humans labeling them in any way, which allows it to use lots of publicly available data, with an automatic process to generate inputs and labels from those texts.

Given the hardware and time constraint, a smaller version of the model has been employed. DistilBERT [3] is a distilled version of the BERT base model. The model is uncased, meaning it does not differentiate between ”english” and ”English.” It is a transformers model, smaller and faster than BERT, which was pre-trained on the same corpus in the same self-supervised fashion, as its parent model. To perform sequence classification the model is provided with a sequence classification/regression head on top (a linear layer on top of the pooled output).

Before feeding data to the model, text is transformed using the Distilbert Tokenizer, a pre-trained tokenizer model.

All the different data pre-processing techniques, before tokenization, have been studied in order to understand model performances. The results for the different combinations are shown in Table II

Models	F1 Score
Base model	0.7932
Stopword	0.7782
Stopword + Lemmatization	0.7726
Stopword + Stemming	0.7656
Data Augmentation	0.6824

TABLE II: Scores with different techniques

As it is possible to see from the table, none of the techniques seems to improve performances. Figure 2 also shows the plots for the train and validation loss. A divergence of the validation loss can be seen, hinting at signs of overfitting. The selected data augmentation technique has not improved the overfitting problem.

V. Model Evaluation

The performances of the selected model evaluated on the test set are summarized in Table III

Models	F1 Score	Accuracy
Base model	0.793	0.789

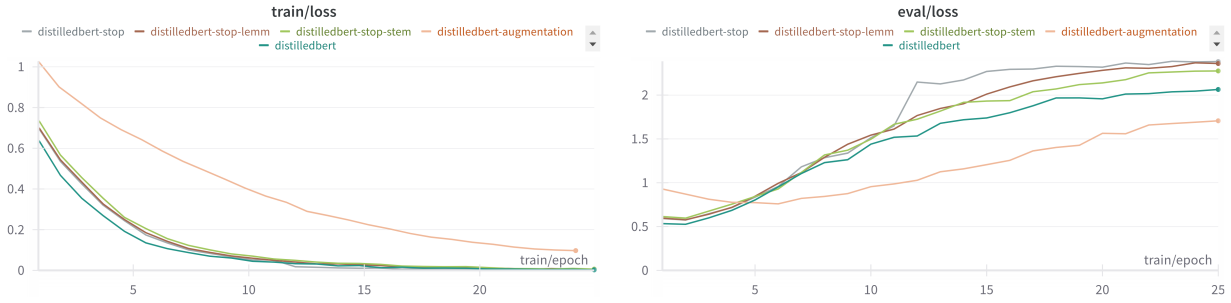
TABLE III: Final scores

The scores are similar to the ones obtained on the validation set.

VI. Experiments

A. LLM-Based data augmentation

This subsection, will present another trial at augmenting the dataset. The idea is to prompt a large language model (LLM) to generate two additional sentences for each sample [1]. These sentences should convey the same meaning and sentiment as the original one. The chosen LLM is *llama-3-8b-Instruct-bnb-4bit*



((a)) Train loss

((b)) Validation Loss

Fig. 2: Loss during train

from the LLaMA family [4]. The prompt provided is presented in the box below.

Prompt

I'll give you a sentence with positive, neutral, or negative feelings, reflecting the sentiment of the user writing it.

For example:
-"I love this ice-cream, it is so good ! :-)" : positive.
-"Last week I visited San Francisco" : neutral.
-"Disappointed by the last Avengers movie..." : negative.

For the sentence, I want you to rephrase it in two different way keeping the same sentiment. The structure has to be:
ANSWER A: answer a
ANSWER B: answer b

Response

ORIGINAL TWEET: Hmm..You can't judge a book by looking at its cover
ANSWER A: You shouldn't make assumptions about something just because it looks a certain way.
ANSWER B: Don't be fooled by appearances; there's more to it than meets the eye.

After injecting the synthetic tweets into the dataset, the model has been retrained and performances on the same validation set have been assessed. The results achieved are the followings:

- F1Score: 0.7527
- Accuracy: 0.7483

The procedure did not improve the scores obtained on the validation set.

B. *Output Attention Mask Interpretation*

The goal of this section is to identify the most important words in the text based on how much attention the model assigns to each token (word). First, the text is tokenized and prepared as input for the model. The model processes the tokenized input and calculates attention weights, which indicate how much focus each token receives from the model during its processing. These attention weights are then summarized through an average on layers and heads to identify the best scoring tokens. These token are considered as the most important because they will play a crucial role in influencing the model's understanding of the text's sentiment. Additionally, special tokens like [CLS], [SEP], and [PAD] have been filtered out before identifying the most important word. This filtering helps to focus the analysis on real words rather than structural or padding tokens. The most important words for each sentiment are reported in Figure 3.

A Jaccard similarity score is computed between the most important tokens and the selected text provided in the dataset. It measures the similarity between two sets of tokens computing their intersection over union. This score ranges from 0 to 1, where 1 indicates perfect similarity and 0 indicates no similarity. Only tokens with attention score inside a 20% range of the highest score are selected. The achieved score is 0.35 pointing out that, given the 20% threshold, the model fails most of the times to identify the entire set of relevant words for classification. Experiment with various thresholds might improve the identification of the most significant words and increase the Jaccard score.

((a)) Most frequent positive words

((b)) Most frequent negative words

Fig. 3: Example of most frequent words

VII. Conclusion

This study develops a sentiment classification model using a dataset of tweets entries with sentiment annotations. Various pre-processing techniques are employed, and DistilBERT, a distilled version of the BERT model, is selected for the classification task. The experiments show that pre-processing techniques such as stop words removal, stemming, and lemmatization do not significantly improve model performance.

The base model achieves an F1 score of 0.7932, indicating effective sentiment classification. However, attempts to address overfitting through data augmentation, using both traditional methods and an LLM for generating additional samples, do not reach better results. This suggests the need for further optimization of augmentation techniques.

The analysis of attention weights provides insights into the most influential words for sentiment classification, highlighting the significance of certain tokens in determining sentiment.

References

[1] Vitor Gaboardi dos Santos, Guto Leoni Santos, Theo Lynn, and Boualem Benatallah. Identifying citizen-related issues from social media using llm-based data augmentation. In *International Conference on Advanced Information Systems Engineering*, pages 531–546. Springer, 2024.

[2] Divya Khyani, BS Siddhartha, NM Niveditha, and BM Divya. An interpretation of lemmatization and stemming in natural language processing. *Journal of University of Shanghai for Science and Technology*, 22(10):350–357, 2021.

[3] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[5] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019.