Technical Report Challenge 3 - Sentiment Analysis

Algorithmic Machine Learning Course (AML), Spring 2023, EURECOM

I. Introduction

Sentiment analysis is a sub-field of natural language processing (NLP). It is a rapidly growing field, as large databases become increasingly available, and companies have an interest in automatically extracting sentiment from internet users. As a matter of fact, with the increasing amount of content and debate on social media platforms such as Twitter, there is an interest in automatically extracting insights from large amounts of unstructured text data and understanding the overall sentiment of a population towards a particular topic. Capturing sentiment in language is important in these times where decisions and reactions are created and updated in seconds. The volume and diversity of tweets call for an automated solution capable of predicting the sentiment behind unseen tweets, given only examples of sentiment-labeled tweets as training data. The dataset employed consists of tweets from Figure Eight's Data for Everyone platform [1]. Two different solutions have been explored: a Naive Bayes baseline model with different granularities of text cleaning, namely light and heavy cleaning [2], and a Robustly Optimized BERT Pre-training Approach (roBERTa). The experiments are aimed at the quantization of the impact of the mentioned pre-processing steps on the model's performance, compared with the Transformer tokenizer, being light cleaning involved with essential steps, whilst heavy cleaning encompasses also more invasive techniques.

II. Data Exploration

The training dataset at hand consists of 24732 unique samples characterized by four features, namely a unique identifier, the tweet text, a selected part of the tweet considered as useful, and the sentiment i.e. the target value. The test dataset consists of 2748 items. The task will exclusively focus on the tweet column in order to predict the sentiment. In Figure 1, the distribution of sentiment values is shown with a slight majority of the tweets being of neutral sentiment.

A histogram reporting the length of tweets from both training and test datasets is shown in Figure 2. Besides the difference in size between the training and test datasets, there is a slight difference between the two distributions. The training dataset shows a higher prevalence of 40-character-length tweets and exhibits a sharper decrease beyond this point, whilst the test one appears to have a roughly uniform distribution with respect to this indicator measure. The length of tweets within the training dataset per each sentiment is shown in Figure 3.

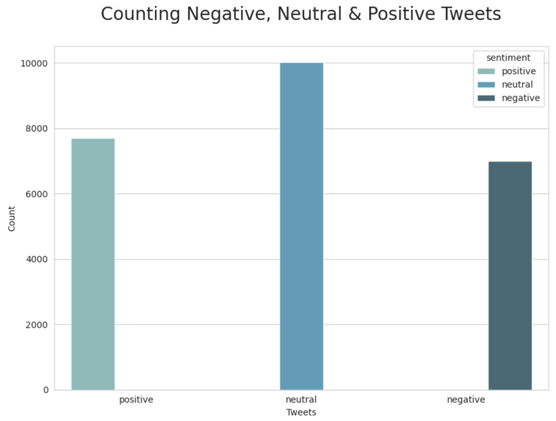

Fig. 1. Number of instances in the training dataset grouped by each sentiment

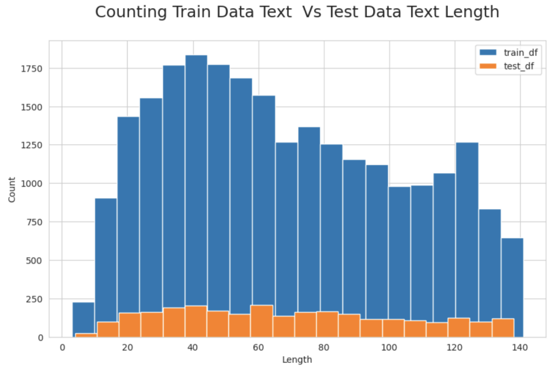

Fig. 2. Number of tweets in the test and training dataset for each specific tweet length

A word cloud per each sentiment has been generated, as shown in Figure 4. By visualizing them, it results that certain words tend to appear more frequently within each sentiment group, underscoring the unique lexical patterns associated with positive, neutral, and negative sentiments. It is safe to say that the highlighted keywords alone lack context to generate accurate predictions.

Meta features Tweets, as the ones in the dataset at disposal, tend to be quite far from plain English text, as they are usually characterized by slang, abbreviations, emojis, emoticons, and misspellings. This motivated the need for ways to learn more about the data and separate it into additional features. An attempt for extracting meta features related to formats has been carried out. Thus, an analysis of the average word length, punctuation, and hashtag frequency per each sentiment has

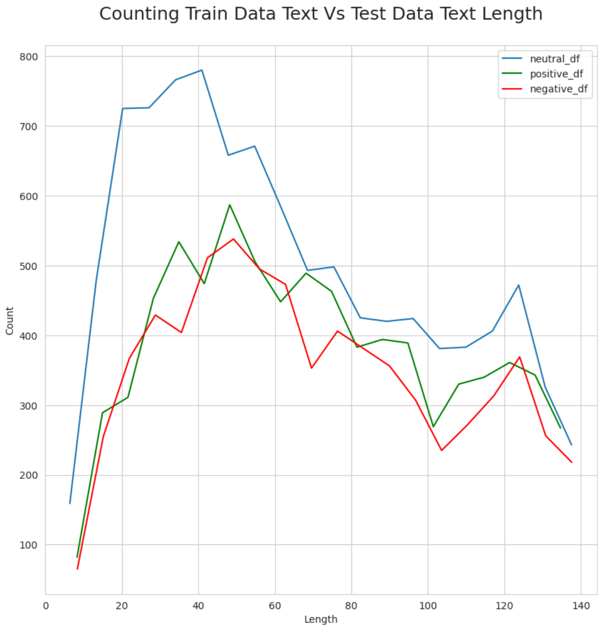

Fig. 3. Length of tweets per sentiment in the training dataset

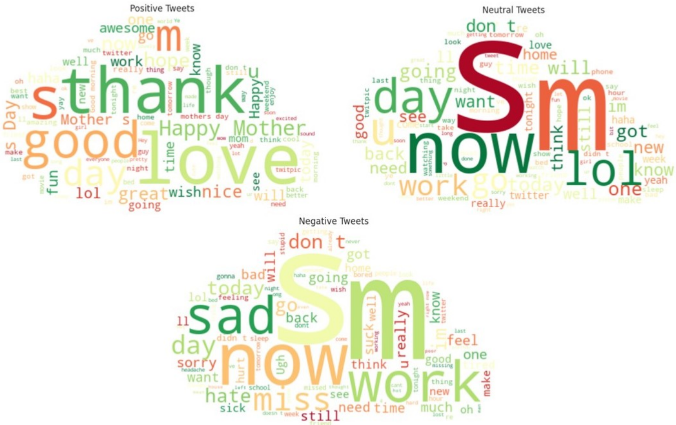

Fig. 4. Wordsclouds grouped by each sentiment with stopwords removed

been performed. The underlying intuition is that the absence of punctuation or minimal use of punctuation combined with a non-short average word length might indicate a more neutral or objective tone, and hence discretize it better from the other two sentiments. Nevertheless, the over-mentioned factors were similarly distributed among the sentiments. For instance, the average word length for each sentiment is shown in Table I. A slight tendency for shorter words to show up in negative tweets can be observed, but this indicator remains still negligible in magnitude. Furthermore, both the number of punctuation symbols used, shown in Table II, and the patterns of these symbols were explored, but no significant results were found. Hence, this information was not used to enrich the models.

Sentiment	Average Word Length
Negative	4.355
Neutral	4.467
Positive	4.533

TABLE I
AVERAGE WORD LENGTH FOR DIFFERENT SENTIMENTS

Tweet Sentiment	Punctuation Frequency
Negative	3.76
Neutral	3.35
Positive	3.66

TABLE II
PUNCTUATION FREQUENCY IN TWEETS BY SENTIMENT

Emojis and Emoticons An exploration into the usage of emojis and emoticons was carried out, since it has been shown in the literature that including emojis when performing social media sentiment analysis leads to robust improvements in the sentiment classification accuracy (See [3]). However, our dataset did not contain any of the emojis contained in the ”EMOJI_DATA” dictionary of the ”emoji” python library [4].

Then, the search for emoticons was conducted, i.e. sequences of keyboard characters used to illustrate a facial expression (or to render some kind of picture or symbol), such as ”:)” for a smile, ”:(” for a frown. The ”EMOTICONS_EMO” dictionary of the ”emot” python library [5] was used to look for emoticons in our corpus, and the findings are presented in Table III. The differences in the average number of emoticons per sentiment suggest that users might use emoticons more often when expressing positive sentiment compared to neutral or negative sentiment. However, the overall use of emoticons was still relatively low in all categories.

In general, devising a method that handles emojis and emoticons in a way that includes them in the sentiment analysis is fundamental and highly advised. Indeed, as is described in the following section on Pre-processing, an algorithm that replaces emoticons with their textual description was implemented as part of a ”heavy” text cleaning procedure. However, it must be noted that, when working with large pre-trained models (e.g. the transformer models described in the ”Models” section), they are often able to interpret correctly such symbolic contractions without the need for any dedicated pre-processing.

Tweet Sentiment	Average Number of Emoticons
Negative	0.018
Neutral	0.016
Positive	0.030

TABLE III
AVERAGE NUMBER OF EMOTICONS FOR DIFFERENT SENTIMENTS

III. PRE-PROCESSING

The ultimate goal is to provide a comparison of the results obtained by using either a less intense (**light**) text pre-processing or a more intense (**heavy**) one before the baseline model and then to compare these outcomes with the ones of a transformer-based model closer to the state-of-the-art. More specifically, light cleaning was charged of performing lowercase conversion and uni-code characters and links removal, while heavy cleaning furtherly removes user handles, hashtags and single letters and numbers surrounded by space. As far as it concerns the *emoticons*, heavy cleaning was charged to convert them into words to later be fed to the chosen model, whilst the less invasive light cleaning left them unchanged. In addition, heavy cleaning exploited a library of words to remove the inflectional endings of words to return them to their base form, which is known as the lemma. This process is carried out via the NLTK library [6], which is used to assign part-of-speech (POS) tags to words in a given sentence or text. Indeed, POS tagging is the process of labeling the words in a text with their respective parts of speech, such as nouns, verbs, adjectives, adverbs, etc. As for roBERTa, the ideal amount of cleaning to perform on tweets is debatable, since these kinds of models are able to infer the context of words provided their placement in a sentence in relation to all other words. Hence, the preservation of such information results to be advised [7]. Therefore, text pre-processing before the transformer model was limited to the aforementioned light cleaning procedure and a tokenization algorithm. This algorithm is a sequence of operations that separates the input text into tokens, encodes them as numbers in relation to a vocabulary handles special tokens and out-of-vocabulary terms, and performs padding or clipping so that all input sequences end up having the same length. This is intended to create appropriate embeddings as a way of including the text data in the model.

IV. Models

For what concerns the models, first a Naive Bayes classifier [8] has been employed on the Bag-of-Words features of the training data. This stands in as a baseline since its basic idea is to find the probabilities of classes assigned to texts by using the joint probabilities of words and classes [9]. To test out the impact of cleaning on the final performance, both light and heavy cleaning have been applied to the over-mentioned model. In the context of sentiment analysis, in which contextualization plays a crucial role, Transformers have been chosen over Long short-term memory (LSTM) networks. Indeed, Transformers, via the self-attention mechanism, can effectively capture long-range dependencies, leading to a more improved representation of the input text than LSTM. Transformers are suitable also for the handling of long sequences of text, making them more appropriate for tasks where the sentiment may be influenced by the overall context [10]. As for the choice of the most suitable Transformer, given that including emoticons in the social media sentiment analysis would robustly improve the sentiment classification accuracy, and BERT-based encoders don't support them, a Twitter-RoBERTa encoder [12] has been employed. This choice was also motivated by the fact that this Transformer has been trained on 58M tweets and fine-tuned for sentiment analysis with the TweetEval benchmark, hence it results to be more suitable for the task at hand.

V. Results

The main metric used for model evaluation was the F1-score, defined as the harmonic mean of the precision and recall, reaching its best value at 1 and worst score at 0 [11].

A. Naive Bayes classifier

The Naive Bayes classifier on the Bag-of-Words features of the training data has been fed with both heavily and lightly cleaned data. As a result, the model achieves an F1 score of 0.6515 with light cleaning and of 0.6450 with the heavy one.

B. roBERTa

The overall F1 score and the accuracy per each sentiment obtained with light cleaning are the following, with the associated hyperparameters that were used for training:

train_batch_size :	4
val_batch_size :	32
lr (AdamW) :	$1e-5$
eps (AdamW) :	$1e-8$
F1 score (validation, last epoch)	0.7923
Sentiment	**Accuracy (validation)**
Neutral	0.7583
Positive	0.8274
Negative	0.792

VI. Bonus task: Sentiment extraction

The final task of this project consisted of determining which words in each tweet helped the model determine its sentiment. The approach employed consisted of fine-tuning a pre-trained transformer (RobertaForQuestionAnswering, see [13]), that is a roBERTa model with a span classification head on top for extractive question-answering tasks like SQuAD (Stanford Question Answering Dataset). The model has a linear layer on top of the hidden-states output to compute span start logits and span end logits, that correspond to the initial and final characters of the answer in the text used as context for answer extraction.

The sentiment extraction task was formulated as a question-answering problem: given a question and a context, the transformer model was trained to find the answer in the text column (the context), considering the following associations:

1) Question: sentiment column (positive, neutral, or negative)
2) Context: *text* column
3) Answer: *selected_text* column

First, the training and test datasets were adapted and made compatible with the Question Answering format. Then, the *deepset/roberta-base-squad2* [13] model was trained on the training set and used to predict the *selected_text* column for the test set. The training was performed using the *simple-transformers* library [14], a tool to train and test transformers models easily. Finally, the performance of such a model was evaluated using the ***Jaccard Index***, which, for ground truth label set y and predicted label set $\hat{y}$, is defined as:

$$J(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|}$$

To apply it to the sentiment extraction task, the selected_text ground truth and the predicted text were treated as sets.

VII. Bonus task results

The average Jaccard score on the test set is: 0.6120256. The training, performed using the *simpletransformers* library for simplicity, was executed with the following hyperparameters:

$$\begin{aligned}\textbf{learning\_rate}:&\quad 5e-5\\ \textbf{num\_train\_epochs}:&\quad 3\\ \textbf{max\_seq\_length}:&\quad 192\\ \textbf{doc\_stride}:&\quad 64\end{aligned}$$

A few examples of the produced predictions are presented in Table IV in the Appendix.

VIII. Conclusions

Text pre-processing is a crucial step in any NLP problem, and it often plays a key role in achieving good performance in tasks like Sentiment Analysis. While the light cleaning algorithm is composed of steps that are the basics when it comes to handling a corpus with unclean text, the work done showed that the heavy pre-processing algorithm, instead, leads to worse performance when employed before the training of the baseline model. This suggests that operations like the lemmatization are probably too ”aggressive” on the input data and lead to loss of information.

Moreover, the usefulness of the emojis/emoticons conversion included in the heavy cleaning algorithm could not be gauged on the dataset at hand because of the lack of such symbols. A scenario that included more diverse tweets, with both emojis and emoticons being used with higher frequency, would have benefited more from it, but more importantly, it would have shown even more how powerful and tailored a transformer-based model like Twitter-roBERTa is.

Finally, while the reported results clearly show that the transformer model achieves better performance than the baseline, a thorough hyperparameter tuning process, which was omitted for lack of time and computational resources, would probably lead to even better outcomes.

References

[1] https://appen.com/what-we-do/#Sourcing

[2] https://towardsdatascience.com/part-1-data-cleaning-does-bert-need-clean-data-6a50c9c6e9fd

[3] https://towardsdatascience.com/emojis-aid-social-media-sentiment-analysis-stop-cleaning-them-out-bb32a1e5fc8e

[4] https://pypi.org/project/emoji/

[5] https://pypi.org/project/emot/

[6] https://www.nltk.org/api/nltk.tag.html

[7] https://towardsdatascience.com/does-bert-need-clean-data-part-2-classification-d29adf9f745a

[8] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

[9] https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html

[10] https://medium.com/@lokaregns/effortless-sentiment-analysis-with-hugging-face-transformers-a-beginners-guide-359b0c8a1787

[11] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[12] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

[13] https://huggingface.co/docs/transformers/model_doc/roberta#transformers.RobertaForQuestionAnswering

[14] https://simpletransformers.ai/docs/qa-model/

APPENDIX

textID	Sentiment	selected_text	selected_text_pred
102f98e5e2	1	Happy Mother‘s Day	happy
033b399113	-1	Sorry for the triple twitter post, was having ...	sorry
c125e29be2	1	thats much better	thats much better
b91e2b0679	-1	tummy ache	aww i have a tummy ache
1a46141274	1	good.	good.

TABLE IV
SENTIMENT TEXT EXTRACTION PREDICTIONS EXAMPLES