An In-depth Analysis of Public Sentiment: Reddit Discourse on the Israel-Palestine Conflict

Mohamed Arafaath
Computer Science and Engineering / 2024
mohamesa@stud.ntnu.no

María Tapia Costa
Computer Science and Engineering / 2024
mariatap@stud.ntnu.no

Abstract

This paper serves as report to the final project of course TDT4310, Text Analytics and Language Understanding. It provides a detailed explanation of the project's structure, techniques and rationale followed by the team. The objective of the project is to perform sentiment analysis on Reddit posts regarding the Israel-Palestine conflict. We used natural language processing techniques, such as sentiment analysis to analyse how the overall sentiment has changed over time, and identify which topics are most frequent in the whole dataset. We implemented the above using a variety of existing algorithms and models like, VADER , LSTM model and LDA. The quick conclusion of this was that VADER produced better results in less time compared to the LSTM model but both can predict sentiment score at a state-of-the-art standard.

1 Introduction

In today's society emotions are key to everything, marketing, sales, politics and more. Being able to analyse successfully the overall sentiment of a review or social media post can uncover hidden information which can then be used as feedback to improve or modify the product or service. However, in politics this is fundamental. Politicians have to be very aware of how their message sinks in for the people, and from there alter their speech or idea to what suites the people best. The focus of this project is to analyse how people have felt and feel today about one of the biggest ongoing world wide conflicts, Israel and Palestine's war. This has been the principal motivation of this project as it can be very interesting to see how the population's point of view has changed as the war advanced.
To fullfill this project's purpose we applied the NLP [2024] techniques learnt in the course and used some time to research about state-of-the-art work carried out by other professionals in the sector. We learnt about sentiment analysis [2023] and its implementations. We figured that, just like in many other NLP tasks, the pre-processing of data is a crucial aspect on the project's results, therefore, spent large amounts of time investigating and reading about models and algorithms used to process data accurately for sentiment analysis.

2 Background

In this section you will be able to find all necessary information about the specifics of the project like technical terms, theory behind the algorithms and models used and brief explanation on why we chose these compared to other existent techniques.

2.1 Pre processing

Aiming for readers to have a better understanding, we believe that it is best to start from the beginning of the process, the objective being you can link steps of the project easily.

Pre-processing in natural language processing is crucial as text in the tech world is a synonym for freedom. It is not easy for computers to take clear and accurate conclusions with raw text, so we have to process and format it in the most fortunate way possible to achieve to get shorter execution time

and more precise results. Pre-processing text includes many steps and you can choose which of those you implement. For VADER, few preprocessing is needed although it would infer longer execution time, so deleting stop words, tokenising and lemmatising the text is not actually needed but has been implemented by the team. However, it does need a superficial clean of the text, like eliminating non-ascii values (emojis), eliminating URLs, and punctuation.
Stop words[Sto (2023)] are those words used in any language to make your sentences and make sense to your paragraphs, such as 'the', 'and' or 'in'. Most of search engines are programmed to ignore these they provide very few information to the phrase.
The process of tokenising involves splitting text into a list of tokens. Tokens[Tok (2023)] are smaller units which form your sentence or paragraph. For example, words would be tokens of your sentence and sentences would be tokens of your paragraph.
E.g. "John likes to ride on a bike" Tokens={ 'likes', 'ride', 'bike'}.
As we know, we can use the same root word to express feelings or events in many situations, past, present, future, talking about yourself or other people. However, the actual meaning of all these variations provide the same information, and here comes the process of lemmatising the words. So, for now, we already have identified and eliminated stop words and identified the tokens. The next step is to lemmatise these tokens to get the root word and analyse it's meaning.
E.g. Tokens={ 'likes', 'ride', 'bike'}. Lemmatised_tok ={ 'like', 'ride', 'bike'}. Notice that token 'likes' has changed to 'like' as this is the root word we can use to explain and describe many other things.

2.2 Vader

VADER [VAD (2023)] is a NLP tool used to analyse the sentiment of a text. It is designed to interpret the specific and general sentiment expressed in written text. Due to its short preprocessing of input needed, it can work out of the box which comes in handy when wanting to analyse text directly without any additional steps. Here is an image representing the flow of preprocessing work previous to the implementation of VADER.
The output of vader sentiment are individual scores, as percentages, for labels 'positive', 'negative' and 'neutral' saying how positive, negative or neutral it believes the sentence is. In addition, it also returns a 'compound' value which determines the overall sentiment by adding the individual values and then normalising it between the two extreme values (-1; extremely negative, 1; extremely positive). This measure is the one we are interested in stating the following:

- positive sentiment : (compound score >= 0.05)
- neutral sentiment : (compound score > -0.05) and (compound score < 0.05)
- negative sentiment : (compound score <= -0.05)

In our specific case, we got a quite balanced out result between the three classes. The image below represents the scores resulting from applying Vader to both training and test dataset. As it is expected, the neutral class has a much lower amount of posts as the ranges stated by Vader's documentation are not exactly balanced.

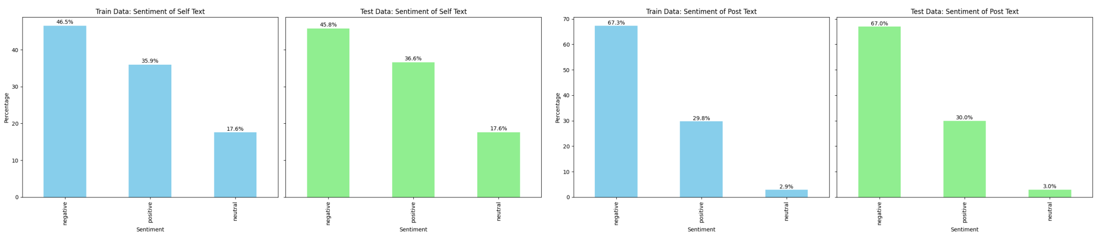

Figure 1: Distribution of classes from Vader sentiment.

2.3 RoBERTa

RoBERTa (2023) is a transformer-based language model that uses self-attention to process input sequences and generate contextualized representations of words in a sentence. Although it seems like quite a definition, let's dive into the meaning of the key terms. A transformer-based language model, as it name infers, is a model which uses transformer architecture, meaning they are well designed to handle sequential data, like text. This means they perform well in understanding the context of words in sentences without the need for recurrent neural networks.

RoBERTa (2024) is built on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates. RoBERTa has been seen to outperform BERT model as it is trained on a much larger dataset, implementing a more effective training providing a more robust model and a broader representation of words. As it is a more complex model, it needs more data manipulation before inputting text into the model. The image below represents how data has been processed before inputting it into the model.

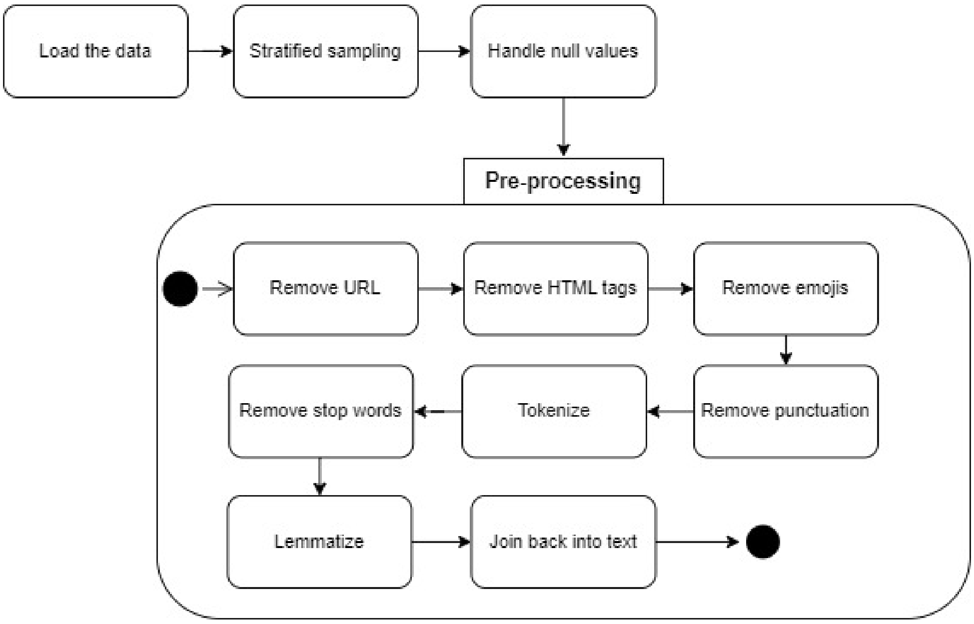

Figure 2: Pre processing before RoBERTa

2.4 GloVe

GloVe (Global Vectors for Word Representation) is a model of distributed word representations. It uses a global corpus containing word to word co-occurrences to generate aggregated word embeddings. Word embeddings are just the way we have to convert text in a format that computer algorithms can understand and process, to perform a variety of tasks like in our case, sentiment analysis.

GloVe has various versions of its model and all very different in size and quality. In our case, we have used the Twitter GloVe file trained on 27B tokens and with 100 vector dimensions. To know more visit (Pennington et al., 2021).

2.5 LSTM

Long Short Term Memory (LSTM, (2023)) model is an improved version of recurrent neural network. These are just machine learning models which use input in a more complex way compared to linear models such as, Logistic Regression, aiming to exploit the input to its fullest leading to more data to train the models and eventually resulting in better results. Although LSTMs are normally aimed at time series and sequences, they are also very good in recognising long term dependencies and have been seen to work quite well with speech recognition (see article from 2022).
The image below represents the work flow and architecture of this machine learning model.Visit either of the references in this section to learn more about this.

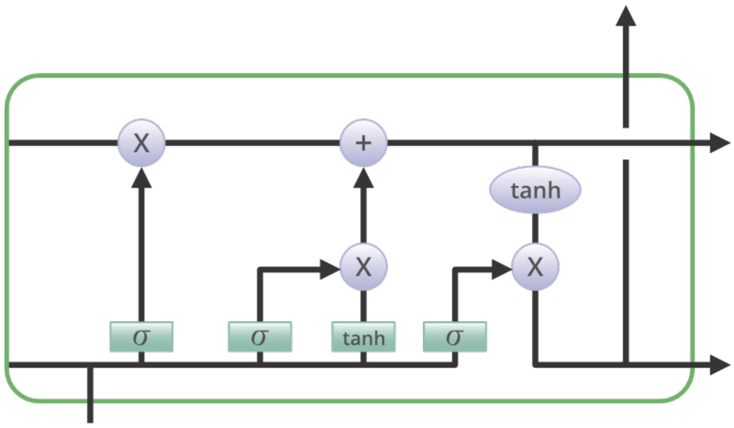

Figure 3: LSTM architecture and work flow

2.6 LDA

Linear Discriminant Analysis (LDA, 2023) is a dimensionality reduction technique primarily utilized in supervised classification problems. It helps classification by modeling distinctions between groups and effectively separating two or more classes. Which is very useful in the scope of the project. Additionally, dimensionality reduction is the process of reducing the number of features (or dimensions) in a dataset while retaining as much information as possible. Taking into consideration the size of our dataset, which will be dived into later on in the paper, this is very beneficial.

This was mainly used in the topic modelling part of the project as it is has been seen to work quite accurately for these process both in class and in state-of-the-art-work such as in this paper by Anupriya and Karpagavalli (2015).

3 Related Work

This section will dive in the research made by the team to gather information of the best models and alternatives to carry out this project to the best of our and today's technology abilities.

To begin with, we began investigating about the different techniques of tokenisation there are nowadays. The article published in Research Gate by Erkan and Gungor (2023) was quite enlightening and served as a baseline for our work. However, taking what we have learnt in lectures and the paper published by the International Conference ACITE by Muthuvelu et al. (2022) was a turning point for us to use the built-in library from nltk, `word_tokenize`, to tokenize the text before feeding it into the VADER model, although, as mentioned before, was not necessary to do so.

Moving onto the next steps of the project, we were very intrigued and motivated by the paper (Alorini et al., 2021) as it displayed a very similar project that we were trying to carry out. Furthermore, the paper offers a detailed explanation of how can LSTMs help and work accurately for NLP tasks when compared to more traditional methods like TextBlob (in this link you can learn more about TextBlob [2019]). In addition, the paper by Catayna and Clariño was useful to tick off NaiveBayes off our list of possible architectures to use as it stands out that LSTM gave better results. It also, gave us a small insight of what would become our biggest adversity throughout the project, the size of the dataset.

The following articles were less useful but not less important for us throughout the project as they helped us to understand the rationale and reasons to choose these techniques rather than other alternatives.

- Elbagir and Yang (2019) paper helped to see VADER being implemented in a similar environment as ours

- Wongkar and Angdresey (2019) paper introduced us to the option of using NaiveBayes model, however we decided to go for LSTM when we found paper mentioned previously in this section.

4 Architecture

This section will detail the final architecture of the project as a whole, from start to end. Understanding, the architecture of a project is key to implement the project again for future improvements. The figure below gives you a brief idea of the workflow of the project.

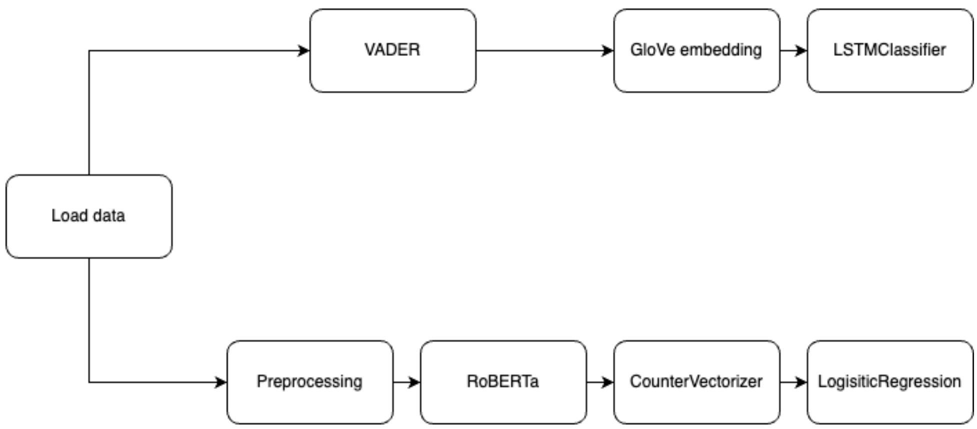

Figure 4: Overview of the projects workflow

4.1 Dataset

Understanding the nature of the dataset in NLP tasks is key, reason to why large part of our work has been dedicated to researching similar datasets and implementing pre-processing techniques to find which suites the dataset best.

An adequate pre-processing of the data can influence significantly the results of our model, that's why getting to know requirements of your model is also crucial. As you could read in this paper, different approaches have been taken depending on whether we were implementing VADER or RoBERTa. As shown in figure 4, the approach taken with both models is completely different. When implementing VADER, we did not pre-process, tokenize or lemmatize the text as state-of-the-art work and documentation state VADER this step is not necessary and would just slow the process down.
On the other hand, as you can see detailed in 2, when implementing RoBERTa, it was precise to implement some pre-processing of text as this impacts significantly the execution time of RoBERTa's sentiment scores.

4.2 LSTMClassifier

This classifier was used to train on the word vectors generated by GloVe, on our un-processed data. In this case, the classifier saw itself somehow impacted with the dimensions of GloVE vectors, as mentioned previously, we used Twitter GloVe model trained on 27 billion tokens with 100 dimensions for each vector. Therefore, it was important to define an LSTM model with corresponding value of `input_size` to the GloVe dimensions.

At first, we implemented a simple LSTM model to have a base reference to accuracy and followed into fine tuning this model to find the best hyperparameters given the nature of the word embeddings. We found that, as we expected, the `input_size` had to be 100, equal to the GloVe dimension. In addition, we found that the optimal size for the hidden layers in LSTM models was of 64 as data was split in batches of 64. Finally, the output layer had to be of dimension 3 as there are 3 possible labels the model should predict.

The image below provides a brief idea of how the LSTM classifier works.

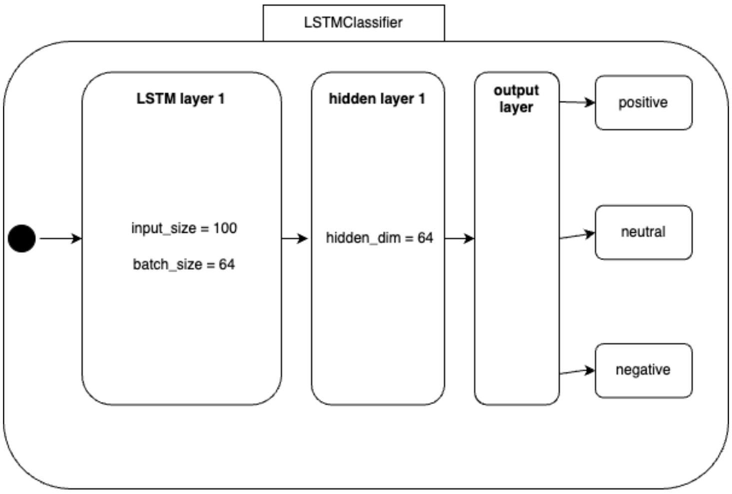

Figure 5: Overview of the LSTMClassifier architecture

4.3 LogisiticRegression

Here you can find an image providing an overview of the LogisiticRegression architecture implemented to predict RoBERTa sentiment labels.

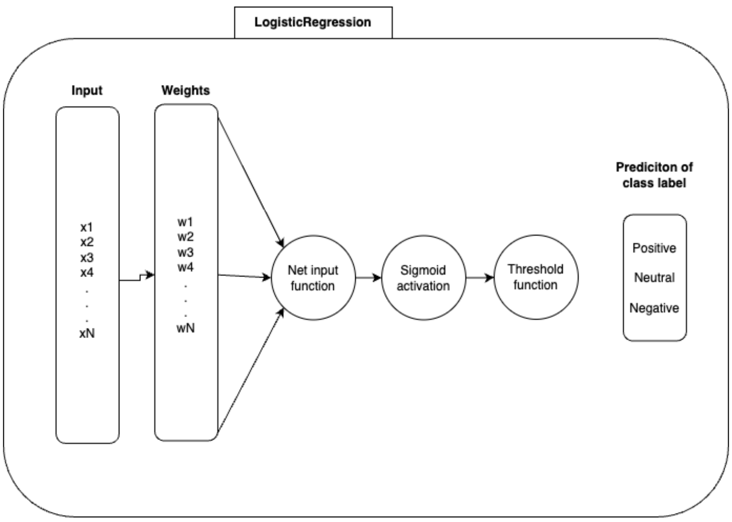

Figure 6: Overview of the LogisticRegression model architecture

5 Experiments and Results

Trying and failing is a major part of research. However, to have a chance of success you need a plan driving the experimental research. So first decide what experiments or series of experiments you plan — and describe them in this section.

5.1 Experimental Setup

In this section you will be able to find relevant information of the project, characteristics, parameters, experiments carried out by the team and more. It is important to understand the difference between labels `self_text` and `post_text` you will see frequently in this section. `self_text` refers to the text users have commented on a post from another user, and `post_text` is the concatenation of both, title and text of the posts Reddit users have posted.

As we have mentioned multiple times in the paper, understanding the dataset was a must for us, and invested large amounts of time in this aspect. When we started, we saw that the formatting of the data was tricky and that the dataset was huge. We found items in each column could have mixed types (finding float types and string types in the same column). In addition, we found that many of the columns included in the dataset were not relevant to the project, so we eliminated them for resource consumption matters.

When it came to pre processing the data we used most of the resources learnt in class such as `nltk, pandas, torch`, with these we were able to implement the cleaning of text, tokenisation and lemmatisation. Moving on, we find the VADER model. In this case we implemented VADER sentiments scores with the `SentimentIntensityAnalyzer` found in the `nltk` library. And, as mentioned previously as well, for the word embedding process, we used the Twitter GloVe file trained on 27 billion tokens and with 100 dimensions.

Finally, when implementing topic modelling we used `LDA` model helped out by `corpora.Dictionary` library which formatted the dictionary and corpus necessary for a reliable result by the `LDA` model. Our `LDA` models were designed to predict on 3 pre-defined topics, **Pro Israel**, **Neutral** or **Pro Palestine** and with a input size of 100.

5.2 Experimental Results

In this section you will be able to visualise all the conclusions we have made throughout the project, including static sentiment distributions, sentiment distributions over time.

5.2.1 VADER and LSTMClassifier

Here you will be able to find the plots created by the team aiming for a better understanding of the results. As you could see in figure 1 the distribution of labels resulting from VADER are more or less as expected with few considered as neutral.

Figures 7 and 8, found below, shows the sentiment distribution over time for comments on users posts and actual text in posts, respectively.

As it is expected, we can observe a big decrease in the amount of neutral posts from users. This is logical as comments don't necessarily give a new opinion on the Israel Palestine conflict but just comment on what that user has said on his/her post.

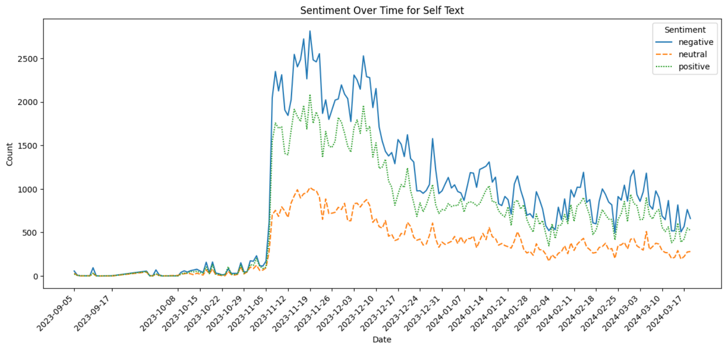

Figure 7: VADER Sentiment distribution over time for comments

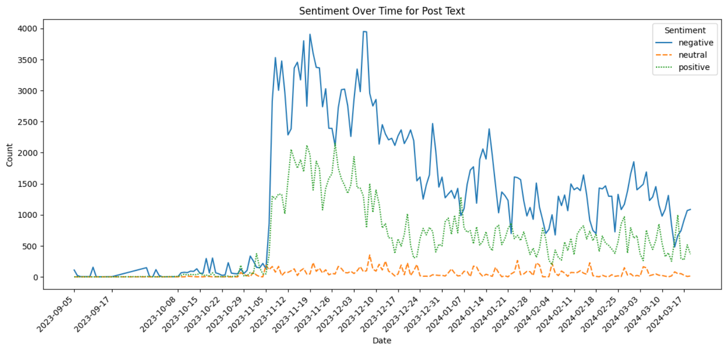

Figure 8: VADER Sentiment distribution over time for post text

5.2.2 RoBERTa and LogisticRegression

Same as for VADER, we have included plots to visualise the sentiment distribution of the senitment scores gotten with the RoBERTa model.Figures 9 and 10, found below, shows the sentiment distribution over time for comments on users posts and actual text in posts, respectively.

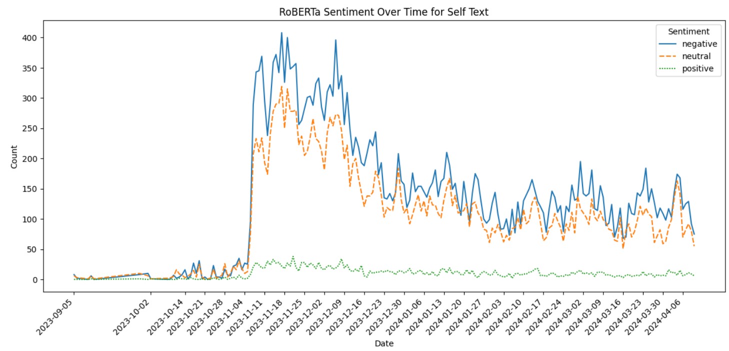

Figure 9: RoBERTa Sentiment distribution over time for comments

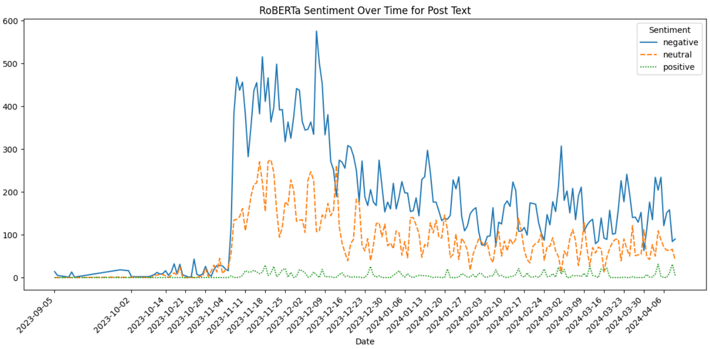

Figure 10: RoBERTa Sentiment distribution over time for post text

5.2.3 Topic Modelling with LDA

Below you will be able to find many representations for the most popular topics (or words) found by our model. To begin with, we would like to show an overview of the results for both `self_text` and `post_text`. As we expect to see, Figure 11 shows that the majority of comments were classified as neutral. Each coloured bar in the graphs; blue, orange and green, represent the amount of comments (or rows) RoBERTa sentiment analyser has considered to be positive, negative or neutral respectiely. It is nice to point out, that in the comments, topic Pro Palestine have a significant number comments classified as negative in comparison to other sentiment labels. However, topic Pro Israel has it more balanced out. In addition, when dealing with text in the actual posts, we can see topic Pro Israel has the most amount of negative classified texts and neutral posts are scarce.

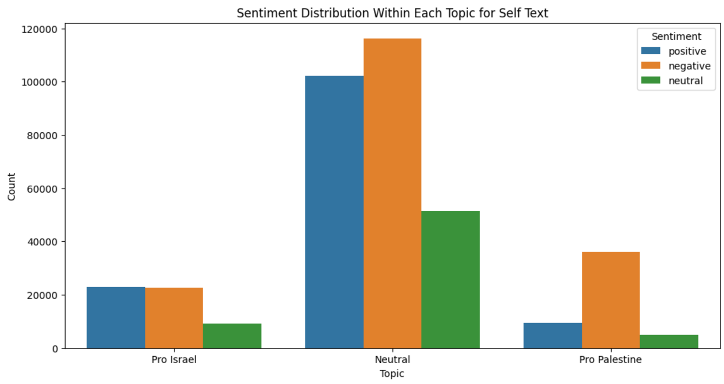

Figure 11: Topic distribution on comments

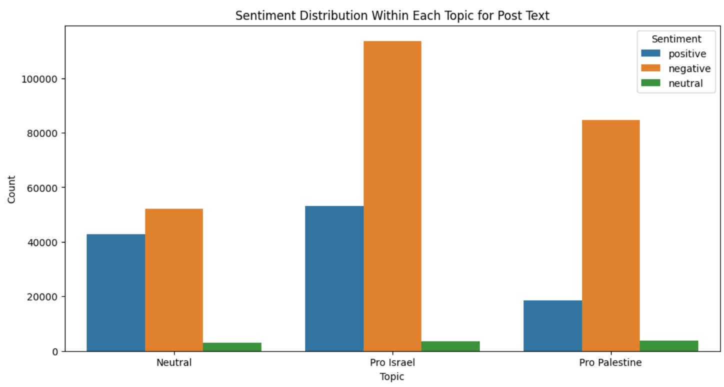

Figure 12: Topic distribution on posts

On the other hand, we also thought it could be interesting to see the distribution of sentiments for each topic over time. Below you can find images comparing both comments and post text sentiment distribution for each topic. Figure 13 shows the sentiment distribution for topic Pro Israel over time, as we are used to see in previous graphs, the amount of neutral sentiments found in the posts is very low compared to the comments text. Also we can identify how the amount of posts considered as positive also decreases significantly, as well as, slight change in the amount of negative texts.

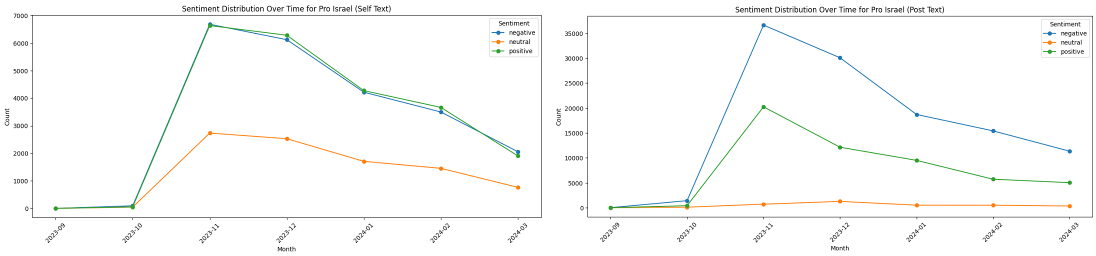

Figure 13: Sentiment distribution over time for topic:Pro Israel

Figure 14 we can see a similar picture as above with a decrease in neutral posts and both negative and positive text vary slightly.

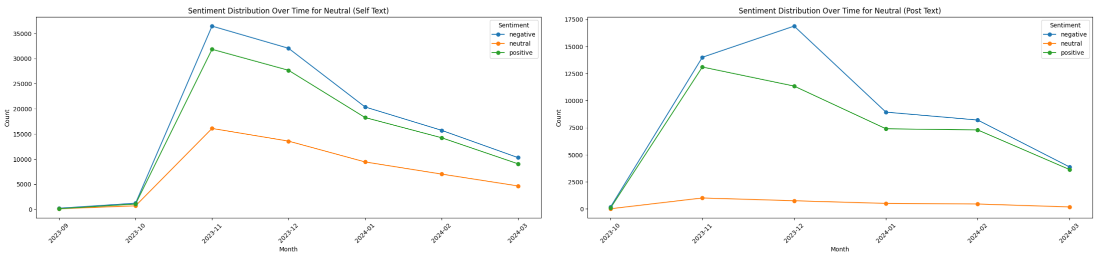

Figure 14: Sentiment distribution over time for topic:Neutral

However, Figure 19 is the most unique image as we can see a fairly low amount of positive comments and posts which is unusual for other topics.

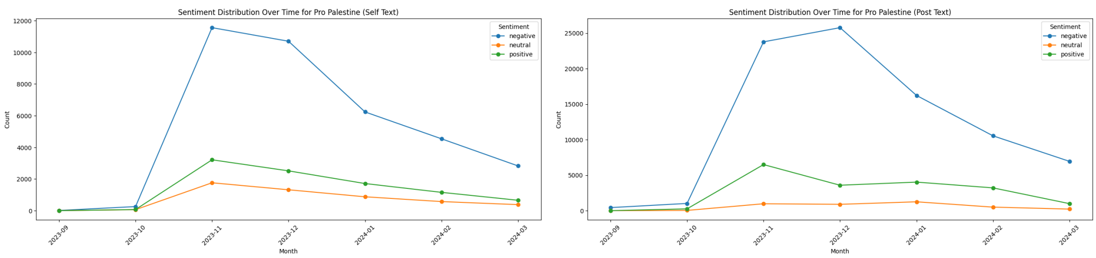

Figure 15: Distribution over time Pro Palestine

As a conclusion, we can see a big change in overall sentiment in time with a high peak of either negative or positive sentiments in the scope of a month between November 2023 and December 2023. Also, we can appreciate a decrease in the amount of negative sentiments towards today's date which is always nice to see.

Finally, we wanted to identify the most common words in each of the sentiments labels.

Figure 16: Positive sentiment

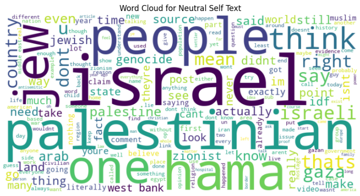

Figure 17: Neutral sentiment

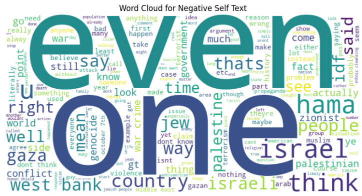

Figure 18: Negative sentiment

Figure 19: Word Cloud for each sentiments

6 Conclusion and Future Work

Code implemented by ProjectRepo

We have enjoyed implementing this idea as we are amazed by how much information you are able to extract when analysing text, that just by reading you are unable to see. This being one of the principal motivations of the project we are also happy to see negative sentiment counts have decreased throughout the course of time and hope this conflict comes to an end soon.

Thinking about the results, we have been quite surprised by the performance of the LogisticRegression model as it outperformed the LSTMClassifier. Below you can see the accuracy of the LSTMClassifier vs the LogisticRegression

- Accuracy for post text: 0.99 vs 0.96
- Accuracy for self text : 0.80 vs 0.68

However, we understand that data inputted into the LSTMClassifier contained much more noise compared to the LogisticRegression model which had clean data. In addition, we would like to remark that sampling the data was very beneficial and reduced significantly the execution time.

Finally, we would like to encourage others to dive into the specifics of the RoBERTa model as it demonstrated to have the best results. This could include the usage of RoBERTa sentiments with Bidirectional LSTMs, known to be very effective in NLP tasks.

References

Textblob. GeekforGeeks, Sep 2019. URL https://www.geeksforgeeks.org/python-textblob-sentiment-method/.

Twitter sentiment analysis using python. GeeksforGeeks, Mar 2023. URL https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/.

Ldadoc. scikit-learn, Apr 2023. URL https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html.

Lstm. GeeksforGeeks, Apr 2023. URL https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.

Robertag4g. GeeksforGeeks, Jan 2023. URL https://www.geeksforgeeks.org/overview-of-roberta-model/.

Stopwords. GeeksforGeeks, Apr 2023. URL https://www.geeksforgeeks.org/removing-stop-words-nltk-python/.

Tokenize. GeeksforGeeks, Apr 2023. URL https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/.

Vadersentimentdoc. GeeksforGeeks, Apr 2023. URL https://vadersentiment.readthedocs.io/en/latest/.

Natural language processing (nlp) tutorial. GeeksforGeeks, Mar 2024. URL https://www.geeksforgeeks.org/natural-language-processing-nlp-tutorial/.

Roberta. Hugging Face, Apr 2024. URL https://huggingface.co/docs/transformers/model_doc/roberta.

Ghaida Alorini, Danda B. Rawat, and Dema Alorini. Lstm-rnn based sentiment analysis to monitor covid-19 opinions using social media data. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021. doi:10.1109/ICC42927.2021.9500897.

P. Anupriya and S. Karpagavalli. Lda based topic modeling of journal abstracts. In *2015 International Conference on Advanced Computing and Communication Systems*, pages 1–5, 2015. doi:10.1109/ICACCS.2015.7324058.

Carlo R. Catayna and Maria Art Antonette D. Clariño. Sentiment analysis and topic classification of twitter-based data set on the face-to-face classes resumption in the philippines during the covid-19 pandemic. In *2022 2nd International Conference in Information and Computing Research (iCORE)*, pages 39–44, 2022. doi:10.1109/iCORE58172.2022.00027.

Shihab Elbagir and Jing Yang. Twitter sentiment analysis using natural language toolkit ... *International Association of Engineers*, Mar 2019. URL https://www.iaeng.org/publication/IMECS2019/IMECS2019_pp12-16.pdf.

Ali Erkan and Tunga Gungor. Analysis of deep learning model combinations and tokenization approaches in sentiment classification. *IEEE Access*, PP:1–1, 01 2023. doi:10.1109/ACCESS.2023.3337354.

Kavitha Muthuvelu, Bharat Naib, Basetty Mallikarjuna, Kavitha Ravindran, and Srinivasan Rajkumar. Sentiment analysis using nlp and machine learning techniques on social media data. pages 112–115, 04 2022. doi:10.1109/ICACITE53722.2022.9823708.

Jane Oruh, Serestina Viriri, and Adekanmi Adegun. Long short-term memory recurrent neural network for automatic speech recognition. *IEEE Access*, 10:30069–30079, 2022. doi:10.1109/ACCESS.2022.3159339.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. Stanford NLP Group, 2021. URL https://nlp.stanford.edu/projects/glove/.

Meylan Wongkar and Apriandy Angdresey. Sentiment analysis using naive bayes algorithm of the data crawler: Twitter. In *2019 Fourth International Conference on Informatics and Computing (ICIC)*, pages 1–5, 2019. doi:10.1109/ICIC47613.2019.8985884.