

LEBANESE UNIVERSITY

FINAL YEAR PROJECT

Tweet Talk: Analyzing Sentiments About ChatGPT

Authors:

James MAHFOUZ
Stephan MAHFOUD

Supervisors:

Dr. Nicole CHALLITA
Mr. Elie DINA

*A thesis submitted in fulfillment of the requirements
for the degree of BSc. in Data Science*

in the

Faculty of Information II

June 21, 2024

Declaration of Authorship

We, James MAHFOUZ, Stephan MAHFOUD, declare that this thesis titled, "Tweet Talk: Analyzing Sentiments About ChatGPT" and the work presented in it are our own. We confirm that:

- This work was done wholly or mainly while in candidature for a research degree at the Lebanese University, Faculty of Information II.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at the Lebanese University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Signed: James Mahfouz, Stephan Mahfoud

Date: June 21, 2024

“Life is like a sewer . . . what you get out of it depends on what you put into it.”

—Tom Lehrer

LEBANESE UNIVERSITY

Abstract

Faculty of Information II

BSc. in Data Science

Tweet Talk: Analyzing Sentiments About ChatGPT

by James MAHFOUZ, Stephan MAHFOUD

This project consists of a sentiment analysis on twitter data about ChatGPT using advanced machine learning techniques. Sentiment analysis is a sub-field of natural language processing, with a goal to predict the writer's sentiment. This study is done using various machine learning and deep learning models to predict the tweets' embedded sentiment. The evaluation of performance of these models prove that a good IAA isn't a requirement to get high performances, good models are not always generalizable to unseen data, and pre-trained Large Language Models are not often generalizable without fine-tuning. The methodology includes data cleaning steps such as removing and modifying unwanted words and URLs, removing stop-words, dealing with contractions, tokenizing, and vectorizing tweets using techniques like TF-IDF and word embeddings. Additionally, the study implements classification algorithms (random forest, maximum entropy and deep neural network), as well as imports Large language models like Zero-Shot and Text Classifications to compare them. Accuracy, precision, recall, and F1-score are the metrics used to evaluate the performance of these models. The results underscore the importance of IAA in the generalizability of the data and not the performance, and the need to fine-tune large language models on the data to achieve reliable sentiment predictions. Overall, this research contributes to the understanding of challenges and limitations associated with sentiment analysis.

Acknowledgements

We would like to thank all those who contributed to the success of our Final Year Project.

First, our thanks go to our professors at the Lebanese University, Faculty of Information II, who helped us gain the necessary skills and knowledge so that we could succeed in this project.

We would like to thank our supervisor, Mr. Elie Dina, a data scientist, for his mentoring, and the time he dedicated for us. Thanks also to his patience and communication skills, we were able to complete our research project seamlessly. He was of invaluable help in the most delicate moments.

We also thank Dr. Nicole Challita, our faculty-assigned supervisor, who guided us thanks to field expertise. Her time management and organizational skills provided structure and guidance at every stage. This allowed us to move through the project with clarity and purpose.

Finally, we would like to thank all the people who advised us and reread this report. This project would not have been possible without the support of each of these individuals.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	iv
1 Introduction	1
1.1 Project Description	1
1.2 Ethical Concerns	2
2 State of the Art	4
3 Data Description	8
3.1 Dataset Acquired	8
3.2 Twitter Specifications	8
3.3 Pre-Processing	9
3.4 Exploratory Data Analysis	11
3.4.1 Understanding the structure of the data:	11
3.4.2 Distribution of Sentiments:	11
3.4.3 Word Frequency Analysis:	13
3.4.4 Distribution of Tweet length:	14
3.4.5 N-gram Analysis:	15
3.5 Inter-Annotator Agreement Process	15
4 Process Followed	17
4.1 Vectorizers	17
4.1.1 TF-IDF	17
4.1.2 Word Embedding	17
4.2 Models	18
4.2.1 Random Forest	18
4.2.2 Maximum Entropy	19
4.2.3 Deep Neural Network	21
4.2.4 Large Language Model	22
4.3 Metrics	23
4.4 Google Form	23
5 Results and Discussion	25
6 Conclusion & Future Studies	30
6.1 Conclusion	30
6.2 Future Studies	31
A IAA guideline	32

B Hyper-parameter Tuning Results	33
C Google Form and Results	42
Bibliography	45

List of Figures

3.1	Distribution of Sentiments in the Dataset	11
3.2	Number of tags/mentions per sentiment	12
3.3	Number of hashtags per sentiment	12
3.4	Word cloud of the 'good' labeled tweets	13
3.5	Word cloud of the 'neutral' labeled tweets	13
3.6	Word cloud of the 'bad' labeled tweets	14
3.7	Distribution of Words Per Emotion	14
3.8	Top 20 bi-grams for sentiments	15
3.9	Confusion Matrix Heat map of annotators and the Dataset labels	16
3.10	Confusion Matrix Heat map between annotators of the Dataset labels	16
4.1	Word Embeddings	18
5.1	RF Confusion Matrix	26
5.2	MaxEnt Confusion Matrix	26
5.3	DNN Confusion Matrix	26
5.4	Comparison of Multi-Class Models	28
C.1	Sentiment Distribution of Responses	44

List of Tables

2.1	Models Performance Comparison	7
5.1	Multi-Class and Binary Classifications Comparison	27
5.2	Predictions on Google Form	29
5.3	Multi-Class Models Performance Comparison	29

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CC0	Creative Commons Zero
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
EDA	Exploratory Data Analysis
ER	Emotion Recognition
GDPR	General Data Protection Regulation
IAA	Inter-Annotator Agreement
LLM	Large Language Model
MaxEnt	Maximum Entropy
ML	Machine Learning
NB	Naive Bayes
NLP	Natural Language Processing
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
SA	Sentiment Analysis
SLM	Small Language Model
SSL	Self Supervised Learning
SVM	Support Vector Machine
TF-IDF	Term Frequency - Inverse Document Frequency

Dedicated to our Friends and Families...

Chapter 1

Introduction

1.1 Project Description

In this rapidly evolving world, public perception towards a product or platform is becoming more and more important. People share their opinions and express sentiments freely on social media platforms, such as Twitter (Rambocas, 2013). This ground of public opinion has people expressing either their support or skepticism towards different products or topics. Therefore, companies are taking advantage of such opportunities in order to understand how a specific population feels about an issue at hand. Nowadays, companies are working on making the customer and employee's experience more enjoyable to increase satisfaction, which would lead to better production and additional consumption.

To be able to reach such goals, analyzing the consumers' and employees' sentiments through handling textual and/or speech data is very important. This process is done by following methods such as Opinion Mining, Sentiment Analysis (SA), Emotion Recognition, etc. SA is a process that analyzes both textual and speech data to extract the emotional undertones of the written message, as well as the spoken language. This type of process, also known as classification, could be either binary, for example: good and bad, positive and negative; or it could be a multi-class classification such as positive, neutral, and negative sentiments (Drus and Khalid, 2019). SA could be done at sentence, document, or feature level.

Nowadays, SA is being wildly used to satisfy consumers' needs in different sectors. It is applied in movie reviews, "financial forecasting, marketing strategies, medicine analysis, and other areas" (Dang, García, and Prieta, 2020).

To be able to handle the large amount of comments and information present nowadays, automating the processes is crucial. Natural Language Processing (NLP) is a form of Artificial Intelligence (AI) that grants computers the capacity to understand a human language through the use of a "natural language". NLP involves statistical and Machine Learning (ML) models which help computer systems to deal with data effectively in all human languages whether written or oral. It can be seen applied in voice-activated digital assistants, translation apps, part-of-speech tagging text generation, etc. (Aqlan, Bairam, and Naik, 2019).

As for the matter at hand, SA is a specific application of NLP that uses techniques such as ML, Natural Language Understanding, and text processing to be able to derive the sentiment that's expressed within a message. Additionally, SA showed a high efficiency, as statistical and ML approaches extract and define sentiment-based content in a text (Aqlan, Bairam, and Naik, 2019), making it the go-to method in the field. For example, if a text contains words such as *love* or *like*, the derived sentiment would be considered as positive. In some cases, sarcasm or irony could be implied which would change the perception towards this matter.

While SA focuses on extracting the polarity in sentiment of a certain text (positive,

neutral, or negative), a complementary task goes further with the analysis: Emotion Recognition (ER). In the realm of NLP, SA and ER are two very interconnected tasks, both offer interesting insights into human communication. Both analyze textual data and emotional undertones. However, unlike SA, ER engages in uncovering the different emotions that the writer of the message is trying to express: happiness, sadness, anger, fear, etc.

In this project, SA will be performed on tweets in the context of ChatGPT, a fast-growing AI tool that was designed to guide and help people with repetitive tasks and offer knowledge. This growth has made it a controversial topic of conversation. Fans of this technology believe that ChatGPT makes their lives easier, while critics are afraid of it, as it is rapidly evolving, leading to a higher probability of unemployment or job deletion. In addition, there exists indifferent people towards the development of these machines, or having mixed feelings. These conversations are taking place on social media platforms, mainly on Facebook and Twitter. The former, even though it is a large social media platform, is not developed for SA because of its unstructured data. However, until 2019, Twitter was mainly used in SA articles by 85% due to its short opinionated tweets (Drus and Khalid, 2019), making it a viable candidate for data collection. By examining these tweets and applying ML models, many insights into the public's views and opinions about this controversial matter will be extracted. This will provide a better grasp as to how ChatGPT has affected people's lives since it was created, and how it is still doing that today.

While working on this project, different hypotheses were considered:

1. Annotated datasets with a bad IAA could give a good performing model.
2. Models could have great performances on a specific dataset but are not generalizable to unseen data.
3. Pre-trained Large Language Models (LLMs) are not always generalizable and need to be fine-tuned.

This project will acquire a dataset and implement the IAA calculations to validate the hypotheses. It will also involve using pre-trained LLMs to compare performances and their generalizability.

1.2 Ethical Concerns

A data scientist primarily focuses on maintaining data quality and model robustness. Nevertheless, the ethical principles under which the work is carried out must also be considered. The General Data Protection Regulation (GDPR)¹ helped to guarantee that the data collection and treatment are to be carried out ethically. This regulation places requirements regarding data collection, consent, and security. In this project, many ethical concerns were raised, spanning from the initial collection of data to the dissemination of findings.

One of the primary ethical concerns that was faced is the GDPR's principle concerning notion of consent. The regulation requires that personal data be collected and "processed lawfully, fairly, and in a transparent manner" with explicit consent from the people involved (GDPR - Chapter 2 - Art. 5 - (a)). However, this project requires the collection of tweets, which are public data by nature. Therefore, obtaining this explicit consent from users for analysis is a challenge. To further complicate

¹<https://gdpr-info.eu/>

the matter, Twitter's consent and user agreement related to third-party usage are buried in the fine print, which makes it less noticeable to the users. Consequently, users may not read or understand these terms, which can impact the ethical considerations about any usage from a third party. To perform the analysis ethically, it was carefully taken into consideration any implications of personal data usage, provided that they may have information such as an address, phone number or such that they would not want to be shared. And so forth, bearing in mind that the user was unaware about the third party usage in many cases, personal information will be omitted to respect their privacy. Furthermore, it should be noted that tweets have high variability, which would not ensure a full anonymization. From Kaggle, a platform for data scientists building AI models with many public datasets available for use, an anonymized dataset was procured. It does not have any features that connect the users of the tweets to the stored data. Additionally, any link to the original tweet that could lead to its author and violate his privacy was removed. This method provided a way to keep the data indefinitely and to ethically analyze it.

Another ethical concern is the bias of the ML models. Since they were trained on social media data, they will accidentally present biases because tweets contain private, cultural, or racial information which will lead to unfair and inaccurate classifications. To be able to deviate from bias, personal information must be anonymized. However, this does not solve the issue, as machines learn from humans, who themselves are biased. For instance, some SA models are racially biased where the classification of a text with the word *black* would give it a negative sentiment, while another with *white* would make it positive.

A third and final ethical concern is also related to another principle of the GDPR: the data minimization and purpose limitation (GDPR - Chapter 2 - Art. 5 - (c)). This regulation means that any personal data collected must only be used for its original purpose. In this project, that purpose is SA. Therefore, to ethically conduct analysis, the data can only be used for its intended research purpose to avoid the misusing and overreaching of the data.

By taking into considerations these ethical concerns and upholding the legal and ethical standards, this project ensures that it does not only comply with relevant regulations, but also respects the privacy and rights of the individuals.

Chapter 2

State of the Art

SA involves using many computational techniques to identify and categorize the emotional undertones in a text, which will be classified into positive, negative, or neutral sentiment. It is widely used to understand the public's opinions and interest as well as improve their satisfaction. By analyzing the data from a text, SA will be the key to gain insight into the needs and preferences of the population studied.

In the 1990s, SA was proposed as a way to detect subjective sentences using computational linguistics. Then in the early 2000s, more research papers started acknowledging and defining SA as a field of study as they demonstrated many ML techniques that classify the text based on sentiment. After the rise of social media in the late 2000s, the increase in public opinion and the surge in data drove more interest in the SA field especially for businesses. Furthermore, this led to greater advancements in ML techniques in the 2010s and the diggings deeper towards more insights for SA (Bordoloi and Biswas, 2023).

Rambocas, 2013 promotes SA as an alternative research technique for collecting and analyzing textual data on the internet. This paper highlights the role that SA has in marketing research because it easily gathers feedback on attitudes without investing in lengthy and costly market activities. Additionally, it showed the whole SA process: from data collection, until the presentation of the output. The fourth and vital step of this process is the sentiment classification which is split into two parts: the first is using ML with models such as Naive Bayes (NB), Support Vector Machine (SVM), or the Maximum Entropy (MaxEnt), which proved to require more resources, whether time, computation, or human! It also created an issue for multilingual translation and the cultural meaning of words. The second technique presented was the use of lexicons. As opposed to the ML classification, it creates a bridge between the language and the knowledge in that language through a list of words in a specific database for each language. The sentiment classification performance achieved more than 80% accuracy on related unseen data for feedbacks having a clear sentiment undertone.

As this field grows, businesses are having their interest piqued towards it. Salinca, 2015, worked on restaurant reviews to classify them on a scale of 5 classes: strong negative, weak negative, neutral, weak positive, and strong positive. The ML algorithms such as NB and Linear Support Vector Classification were shown to have high accuracies in the business review classifications. The aspect-based method proposed involved six steps: the pre-processing, aspect extraction, aspect categorization, sentiment classification, opinion structure generation, and rating calculation. Aspect extraction and categorization involves finding words related to the topic's areas of interest and then categorize them into those sections. In the reviews, the researchers were looking for aspects in specific aspects such as the food, service, and prices. The

paper focused on the method that combines aspect based feature selection and ranking (from strong negative to strong positive) as well as ML algorithms for classifications such as SVM and NB. For the token classification, they achieved high precision and recall as well as a high accuracy of 88.48%. However, they found some misclassifications due to infrequent vocabulary in the training data or sentences that contain specific names or terms. Based on their study of restaurant reviews, their method of combining the aspect-based feature selection with ML algorithms had some improvements in terms of accuracy, precision and recall when comparing them with the SVM or NB methods alone.

Devika, Sunitha, and Ganesh, 2016, compares different approaches for SA to categorize reviews efficiently. These reviews were found on the Internet from social media such as blogs, and wiki. The ML approach, the rule-based approach, and the lexical approach. The latter assumes that the polarity of a sentence or document is the sum of polarities of individual phrases or words by having a sentiment analysis dictionary. By weighting the words, the text will shift to the polar emotions. It uses unsupervised learning and needs highly powerful linguistic sources which pose a problem because they aren't always available. As for the rule-based approach, even though it has a high performance accuracy, it depends greatly on the rules defined for getting the opinion for tokenization. This approach uses a database of words that have a sentiment assigned to them which are used to rate the words and get the polarity score of the sentence. For example, a sentence having the word "love" will have a +1 score, and then this output is checked by the rule: if the score is positive, then the sentence has a positive sentiment. The third approach, the ML approach, proved to be the best because it can work with unknown data after training as well as leading to a powerful classification accuracy. However, nowadays, many of the issues presented in this research paper were mitigated. One of the issues was solved by using the pre-trained embeddings such as word embeddings which allowed the model to be more aware of the context present instead of relying only on pre-defined linguistic databases. For instance, the word embedding using Word2Vec for:

- *happy*: [0.52, 0.87, ..., 0.49]
- *joyful*: [0.50, 0.85, ..., 0.47]

shows obvious similarities which will ease the classification process. This mitigated the issue of unseen vocabulary seen in the various previous researches. Additionally, to improve the rule based approach, NLP technologies have evolved which implemented the refining of the rule-definition process and adaptation to new patterns.

After identifying that the ML approach is the most effective for SA, Ahmad et al., 2017, differentiated between the ML techniques that can be used. Their goal was to identify each one's importance as well as their features and accuracies. Starting with the feature-based model: the MaxEnt, which has a 72.6% average accuracy. What is favorable about this model is the handling of overlapping features. For example, using bi-grams and phrases will not cause any feature overlapping as opposed to other models like the NB. Bi-grams are two consecutive words extracted from a text to analyze language patterns: *I love pizza* has two different bi-grams: *I love* and *love pizza*. Then, they identified the ensemble learning model that works with classification trees: the Random Forest (RF). The benefits of this model is the improved prediction power by using multiple classifiers as compared to the original classifier. This model achieved the highest performance out of all ML models with a 88.39% accuracy. The lowest performing model is the SailAil Sentiment Analyzer, or SASA, with an accuracy of 64.9%. It is based on the NB model and the unigram feature, and

it uses tokenization for feature calculations. The NB binary model was also identified with a 69% average accuracy between its product and movie classification. It is easily interpretable, and assumes independence among predictors based on Baye's theorem which may prove to be a challenge because the algorithm may not be always valid. The multinomial NB also achieved good results. The last model being the SVM proved to outperform the NB and MaxEnt in terms of results with an average accuracy of 80.3%. It contains extensions such as Multi-Class SVM which allows it to work with many classes instead of the default binary.

Reaching the late 2010's, Aqlan, Bairam, and Naik, 2019, published a paper regarding the techniques, concepts and approaches of SA. They identified two main approaches. The first one being the lexicon based approach which focuses on working on a list with opinion words and finding other ideas from the words in a large corpus. It could be done either using a statistical test of randomization, or a semantic approach that gives a value to the sentiment and calculates the similarity of different words. The second main approach is the ML which uses classification models for SA. These models include the NB, MaxEnt, SVM, Decision Trees classifiers, etc. However, a third, more effective, approach for SA was identified which is the hybrid approach. It combines the computational techniques and technologies to yield better results than individual methods. This paper also introduced that Big Data using Hadoop and some big data techniques will be more effective due to its easy handling of unstructured data.

Drus and Khalid, 2019, is a review of the different studies about SA about the methods and tools they used. Just like previous papers, it split the methods into two approaches, the ML approach, and the lexicon-based approach. They found the latter to have a drawback in classification since it is not considering any sarcasm or slang implied. It most commonly uses SentiWordnet and term frequency-inverse document frequency (TF-IDF) and is best used when the data is small. While the ML's common models are the NB and the SVM which are used with bigger amount of data. Comparing the two models, they found that NB is best used when working with a well-formed corpus and the SVM when the dataset has a low shape. After the systemic review of the SA studies, it was found that each approach has its application and they have similar accuracies. As a best practice, they recommended the combination of both approaches, the lexicon-based sentiment scoring function and the NB multinomial model, for the best performance and efficiency.

Dang, García, and Prieta, 2020, highlights the use of Deep Learning (DL) in SA because it solves the issues that NLP usually presents in terms of reduced accuracy and performance in regular ML models. DL offered an automatic learning of features which resulted in better performances. A study proved that the Convolutional Neural Network (CNN), and the Recurrent Neural Network (RNN) models can overcome shortcoming of short text. Another showed that Long Short-Term Memory (LSTM) is very efficient when on different text levels. This paper also shows a comparison of studies by using two approaches for pre-processing: the word embedding and the TF-IDF that uses the sci-kit vectorizer class. The comparative study was done using the Deep Neural Network (DNN), the CNN, and RNN. The results showed that using word embedding as a preparing step is better than TF-IDF because it doesn't show a difference in performance between models, however it affects greatly the processing times (almost faster by double for CNN and RNN with word embedding, and three times faster with DNN). But the TF-IDF show a great difference especially by showing a fluctuation in the RNN model, it shows a low accuracy of 50% in this case. When computational costs should be reduced, they found

that the use of DNN and CNN models is better since RNN has a time-consuming algorithm. As for the performances of each model, DNN has an average accuracy around 75-80% in all datasets. The CNN model is a little bit slower than DNN but still pretty fast with an accuracy of over 80%. As for the RNN model, when word embedding is applied, it has shown a high reliability even though its computationally consuming. However, a slightly higher solidity is observed in this model than the CNN.

TABLE 2.1: Models Performance Comparison

Paper	Models Used	Accuracy	Precision	Recall
Salinca, 2015	Aspect-Based	83.5%	NA	NA
	NB	75.50%		
	SVM	80.34%		
Ahmad et al., 2017	MaxEnt	72.6%	NA	NA
	RF	88.39%		
	SailAil	64.9%		
	NB	69%		
	SVM	80.3%		
Dang, García, and Prieta, 2020	DNN	75-80%	80%	80%
	CNN	80%	80%	81%
	RNN	83%	83%	83%

In this project, the models chosen were the MaxEnt, the RF, and the DNN with word embedding from the TensorFlow keras module because they are very effective in handling textual data in SA. The MaxEnt model, even though it showed to have an average accuracy of 72.6%, was selected over the NB model because it can manage the phrases in the tweets without feature overlapping. The RF technique was chosen because it has a very high prediction accuracy with a 88.39%. It also uses multiple classifiers which in turn will enhance the accuracy, opposed to only using one single classifier. The third model picked is the DL model: DNN. The comparative studies proved that it is very useful to overcome the limitations of traditional NLP methods. Additionally, the DNN model uses the word embedding preparing step instead of the TF-IDF vectorizer since it significantly increased the processing speed while still maintaining almost the same performance (for the whole dataset, it was shown to be three times faster with word embedding with approximately 4 minutes, instead of 12 minutes with the TF-IDF). With an average accuracy of 75-80%, this model provides a good performance all the while having good computational cost, which makes it a viable choice for a large dataset such as tweets. Finally, this project will implement the text classification and zero-shot classification LLMs from HuggingFace to compare them with the built models, as well as testing the hypothesis concerning the generalizability of LLMs.

Chapter 3

Data Description

3.1 Dataset Acquired

The dataset acquired for this study was extracted from Kaggle¹. This platform is known to be an online community for data scientists and ML engineers who develop AI models and solve problems. It is filled with public datasets on a wide variety of topics that can be used.

This dataset was chosen for its relevance to this project as well as its large collection of tweets on ChatGPT. This offered diversity in user interactions and a better understanding of the public's interactions and sentiments towards this innovative technology which might allow for more efficient analysis and model building.

It is also important to consider any ethical and licensing terms associated with the dataset obtained. The creator of the latter applied the Creative Commons Zero (CC0) license, a public domain dedication tool that allows him to waive any related rights to his work worldwide under copyright law. Under the CC0 license, the data can be used, distributed, and worked on, without asking for permission from the creator. Once ethical guidelines and the CC0 license were understood, the use of this dataset will be done with the best practices and as transparently as possible in this project.

The main purpose of this project is to use this dataset of tweets about ChatGPT to calculate the IAA and build classification models, in order to predict the classification of new tweets into one of three sentiment classes (good, bad, neutral). The performance of the models is evaluated using different metrics.

3.2 Twitter Specifications

Twitter is a social network that is mainly used to share thoughts and opinions on topics of interest.

These thoughts are formed in short messages, limited to 280 characters. According to the user's preferences, they can be shared with everyone if the account is public, or to a specific group of people that follow the user when the account is private. In addition, Twitter users, or Twitterers, can mention each other in their tweets using the '@' symbol. This notifies the mentioned user and allows for further interactions. Twitter users can also use hashtags (#) that help categorize the content in the message to make it more accessible to other people interested in the topic. They help in grouping tweets by similar subjects which will improve the exposure of posts and enable the algorithm to more effectively identify popular topics and give users more personalized content that is relevant to them. Hashtags are simple keywords or small phrases which can help provide more details about the actual sentiments. For instance, a seemingly positive tweet containing #hate or #detest will lead to a

¹<https://www.kaggle.com/datasets/charunisa/chatgpt-sentiment-analysis/data?select=file.csv>

correct classification of the emotional undertone of the message, which in this case would be negative (*I love school #hate* would be considered as a negative comment). Furthermore, special characters such as dollar signs(\$), asterisks (*), parentheses (()), etc. can be used.

This project will focus on tweets where users shared their opinions about ChatGPT. SA will prove to be challenging due to the many factors of a tweet. One of them is the use of informal English, including: slangs ("Slaps" means in Twitter lingo as something that is really good), abbreviations ("TBH" is an abbreviation of *To Be Honest*), and emojis (icons that carry emotion). These characters and words might make it more difficult for the ML model to accurately identify the sentiment behind a certain tweet. Additionally, the language is dynamic, always changing, which poses a problem for any pre-trained embeddings and LLMs, thus there is a need to use Self-Supervised Learning (SSL) for such tasks.

Another challenge faced is the addition of pictures such as memes - humorous pictures or short videos that may be captioned with a funny text and spread over the Internet - to the tweets. In the dataset, this element takes form of picture links that are useless to the model. Since analyzing each entry would create more complex data, creating the need for a higher computational power, these links were removed. The last factor considered is the replies of Twitterers. This type of tweets could have nested sentiments where users are expressing feelings directed to specific people or conversations below and not the topic of interest. Since the dataset extracted was anonymized for ethical reasons, it is not possible to take into consideration the context of the tweets and give more precision to the model. Conversely, the latter will take the tweet as it is, without context clues, which may affect its final results.

3.3 Pre-Processing

The role of pre-processing is really important for the success of any data-related project, especially in NLP, such as SA. It is crucial since even small inconsistencies in the dataset can lead to misinterpretations by the model which will skew the results of its predictions. Therefore, pre-processing the data before fitting it into the ML model helps ensure its accuracy. This process aims to transform text data into a format more appropriate for analysis. Pre-processing is based on the concept of "Garbage In, Garbage Out" which emphasizes that the quality of results relies on the quality of input. In other words, taking poor-quality data as input will only lead to a faulty and inaccurate output. To achieve high-quality data, the removal of 'garbage' input is a necessity. It includes noise, or elements that don't contribute to the sentiment expressed, such as links, special characters, and non-textual data. Additionally, it is achieved by handling the linguistic nuances in text data such as spell checking, as well as normalization and standardization. This data preparing step will increase the quality of the model.

The pre-processing includes four main stages: data cleaning, data integration, data transformation, and data reduction. However, in this SA project, only the data cleaning step was used to help prepare the input for later analysis. The primary data focused on will be the tweets that were written. Since the model will be built and trained around this 'tweets' feature, making sure that the message is clean is vital.

- **Data Standardization:** specifically converting the text into lowercase. Since tweets have different capitalization styles, performing this conversion will ensure that the data is consistent and helps to avoid any case variations of the same word. This pre-processing step will also help in the later tokenization

of the tweets. For example, words like 'good' and 'Good' will be treated as two different tokens instead of one which will affect the analysis. However, an issue was created due to the full text capitalization which may indicate a sentiment, excitement or anger (good or bad). To keep this important information, the word "SC", or sentence capital, was added to the lower-cased tweet to indicate that it was originally full upper-cased.

- ChatGPT handling: people write "chatgpt" in many ways. Consequently, it will lead to the creation of unnecessary tokens. Therefore, all instances and different variations of "chatgpt" were transformed into a single occurrence.
- Punctuation handling: The removal of any unnecessary punctuation from the text is the next step. Similarly to the lower-casing, doing this will lead to a more efficient tokenization. Points, commas or other punctuation will lead to additional unnecessary tokens such as 'chatgpt' and 'chatgpt.' which will be considered as two. However, some punctuation like the question and exclamation marks indicate sentiment, which is why they were excluded from the removal.
- Special Characters removal: were later removed to focus on more expressive words that convey sentiments. In twitter, all symbols such as ampersand(&), percent(%), backslashes(/), etc. are accepted. Therefore, to avoid these elements that may create noise, the analysis will be more efficient and reliable.
- Duplicates removal: the removal of duplicate records will not only ensure that all entries are unique and reliable, but will also lead to unbiased insights and models.
- Twitter elements removal: Many components of a tweet can skew with the analysis, this entails elements such as links, mentions or tags, and email addresses. Since they don't carry any sentiment and will potentially lead to inaccuracies, these elements were identified and removed.
- Contractions handling: All contractions were expanded into their base form by using the contractions library. For example, contractions like "I'm" and "Haven't" would be expanded into "I am" and "Have not" respectively. As a result, this expansion will give more clarity to the text and help with the consistency of processing.
- Stop-words handling: To further improve textual data, stop-words like "the", "is", "at", etc. that only provide grammar to the sentence were eliminated. This will help in the reduction of the tweet's size and will enhance the efficiency of the model. However, the negative stop-words such as "not", "nor", "neither", etc. will prove to be useful in the analysis for negative sentiments.
- Empty tweets removal: the elimination of all null values and the empty tweets that were a result of pre-processing will lead to a more efficient analysis. These records aren't relevant and have a big toll on the computational resources. Consequently, it would have led to slower processing and an inaccurate analysis.
- ChatGPT tweets handling: After pre-processing, many rows contained only the word "chatgpt" in them. These tweets will create a biased model towards this word. That being the case, any tweets that solely had that word were removed from the dataset.

- Country handling: some tweets contained the country of users which affects their anonymity. Thus, any country instance was transformed into the word "CNT".
- Tokenization: a tokenized column of tweets was added to the dataset. Tokenization is the process of splitting the text into individual words, in this case called tokens. This step will result in performing many text-related tasks like counting the word frequencies, generate word clouds, and potentially training a better ML model.

These pre-processing steps contributed to preparing the data in a standardized clean format, and were crucial for the beginning of the next step of Exploratory Data Analysis.

3.4 Exploratory Data Analysis

A vital step towards building a model is Exploratory Data Analysis (EDA), as it provides an insight about the dataset. This process is aimed at understanding the latter's characteristics, features, and its structure. It will also help in spotting the important patterns, as well as keeping the data clean. EDA often results in a statistical summary as well as visualizations through graphs.

As for this project, EDA is very necessary. By understanding the distribution of the data across the three sentiments, it will uncover any biases that might exist. For example, data that is skewed towards the negative sentiments might affect how the model is trained. This will ultimately lead to an inaccurate classification of the other classes. Additionally, EDA will also give a better understanding of the tweets by observing words associated with different sentiments.

3.4.1 Understanding the structure of the data:

The dataset consists of 3 columns and 219,294 rows. Each row represents a different tweet. As for the columns: *ID*, the primary key of the dataset, is a unique value for every different tweet. The second column *Tweets* represents the textual tweets written by Twitter users without taking into consideration any attached media such as photos and/or videos. The third one *Labels*, representing the target feature, denotes the sentiment associated with each tweet as either good (positive), neutral, and bad (negative).

3.4.2 Distribution of Sentiments:

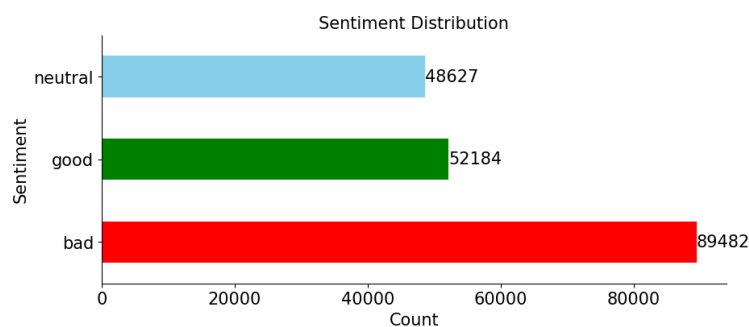


FIGURE 3.1: Distribution of Sentiments in the Dataset

As seen in the figure 3.1, the dataset comprises of 3 sentiment categories: neutral, good (positive) and bad (negative). Approximately, half of the data is labeled as 'bad', while the other is roughly evenly distributed between the 'good' and 'neutral' sentiments. Because of the omission of the user demographics, the bias towards the 'bad' sentiment cannot be known. However, it could be caused by a multitude of reasons such as the rising fear of AI, or the older age groups that take comfort in more traditional approaches rather than the evolving technologies, etc. In addition, the number of tags per emotion will help assess any class imbalance and data composition.

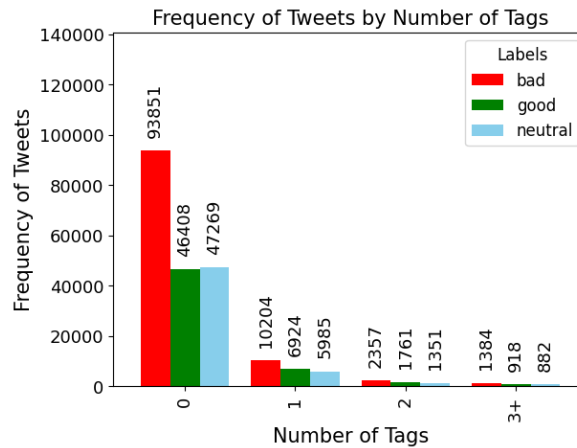


FIGURE 3.2: Number of tags/mentions per sentiment

The distribution of mentions in a tweet is shown in the above figure 3.2. Tweets are usually shown to have no tags in them, regardless of the sentiment. However, it can be seen that the 'bad'-labeled tweets have much more tags than the other two classes. Still, this difference does not show a bias because of the already existing gap between the number of tweets of those 2 groups.

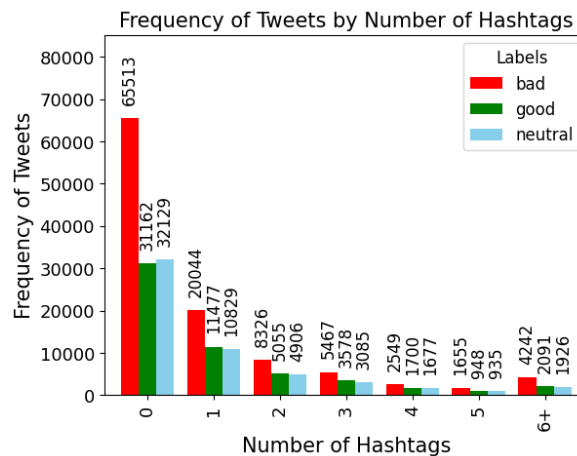


FIGURE 3.3: Number of hashtags per sentiment

The graph in figure 3.3, shows how many hashtags are present in the tweets. Similarly to the previous visual, the 'bad' labeled tweets have more hashtags in them than the 'good' and 'bad' sentiments due to the pre-existing gap of tweets per sentiment.

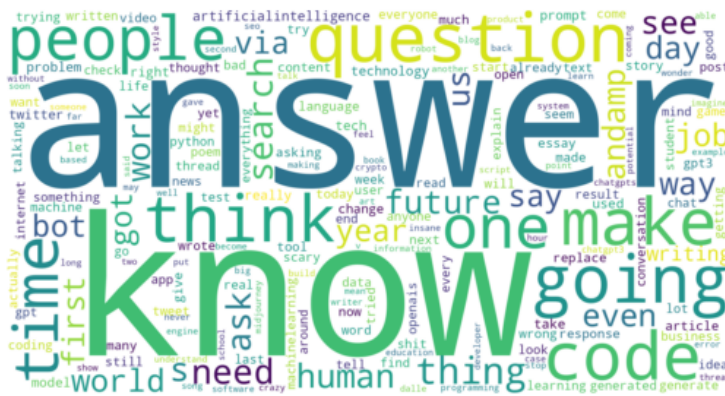


FIGURE 3.6: Word cloud of the 'bad' labeled tweets

it is clear that there is an absence of any words denoting positivity. This absence underscores the lack of positive sentiments in the 'bad' category as well as confirming its negativity.

3.4.4 Distribution of Tweet length:

This EDA technique is about analyzing the distribution of tweet lengths, in terms of word count, for each sentiments category. It will check if there are any noticeable differences in the length of tweets across sentiments.

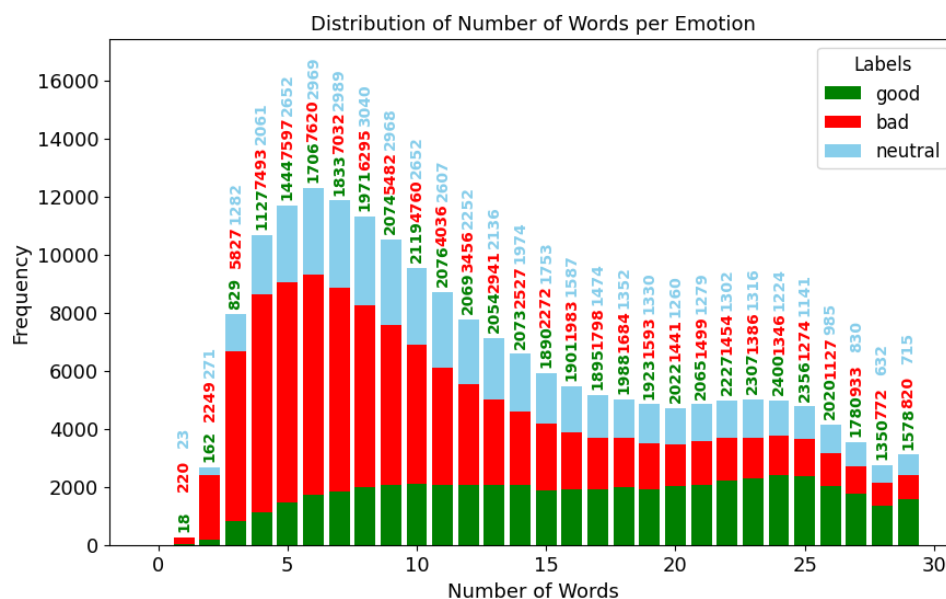


FIGURE 3.7: Distribution of Words Per Emotion

The above figure 3.7 displays the distribution of the number of words used to express emotions: x represents the number of words, and y represents their tweet frequency. One can notice that most of the tweets are between 3 and 13 words approximately. Additionally, it shows that the 'bad' sentiment is mainly expressed with lesser words than the others: it peaks greatly with 7620 tweets containing 6 words and then gradually decreases. As for the 'good' emotion, it is clear that its expressed in a higher number of words.

3.4.5 N-gram Analysis:

This step will focus on the exploration of the distribution of n-grams (sequence of N-words) across different sentiment categories. Analyzing bi-grams can be effective in revealing common words associated with each other for different polar sentiments.

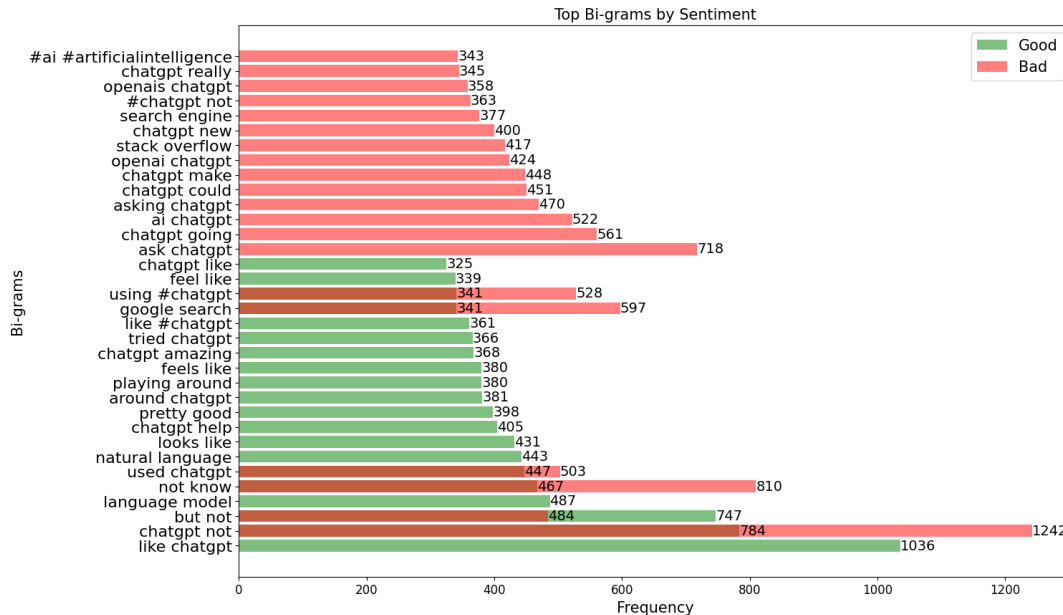


FIGURE 3.8: Top 20 bi-grams for sentiments

The horizontal bar plot shown in figure 3.8 represents the 20 most common two-word combinations in the two sentiment categories: the 'good' labeled tweets are shown as the green bars, while the red ones correspond to the 'bad' labeled tweets. In order to enhance this visualization, less significant combinations such as 'chatGPT and write', 'I am', etc., have been removed. Upon initial inspection, the positive words in the 'good' labeled tweets are instantly noticeable where 'pretty good' and 'chatgpt amazing' take place in the top 20 combinations. As for the 'bad' labeled tweets, while they exhibit less overt negativity, some bi-grams can still be noticed such as 'not know' and 'chatgpt not'. Additionally, the combination 'chatgpt not' was the most frequent words combination with 1242 occurrences in the 'bad' labeled tweets, which shows a lot of negative feeling towards chatGPT.

3.5 Inter-Annotator Agreement Process

In SA, the Inter-Annotator Agreement (IAA) process plays an important role in determining the consistency and reliability of the labeled data. This step is done by having two or more people annotate the dataset's target label according to a guideline and then compare the results with the actual labels. Furthermore, it shows the similarity among the annotations made by different annotators on the same data. Additionally, IAA shows how well they follow the same guidelines, criteria as well as the standards for labeling the data. The heat maps below represents the result labels of annotators compared to the main dataset labels (3.9) as well as a comparison between the annotators (3.10) following a shared guideline (Appendix A).

In the heat maps of the figure 3.9, the x-axis represents the labels by the annotators and the y-axis represents the true labels. In this project, an IAA score of 0.345

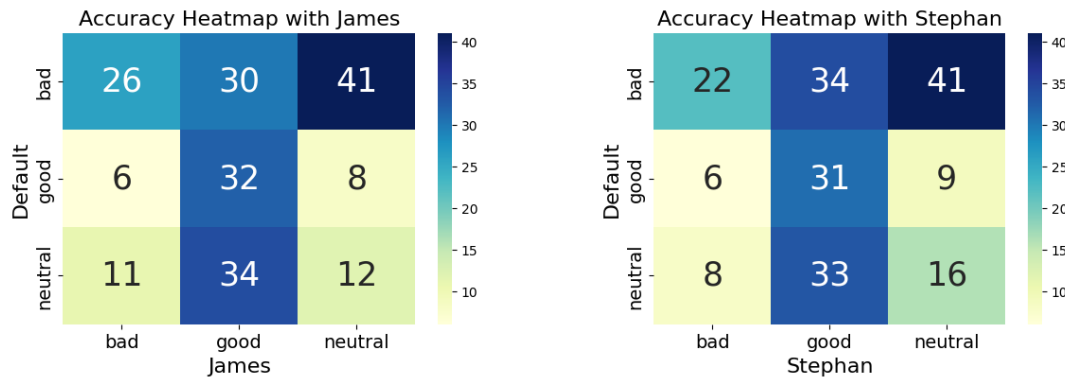


FIGURE 3.9: Confusion Matrix Heat map of annotators and the Dataset labels

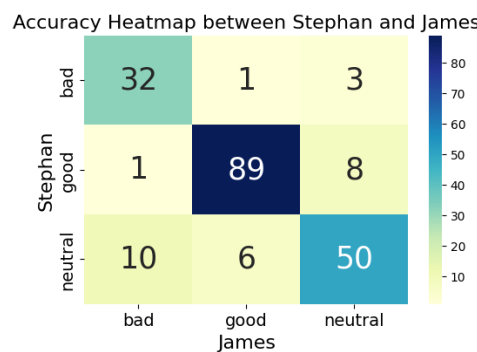


FIGURE 3.10: Confusion Matrix Heat map between annotators of the Dataset labels

with Stephan and 0.35 with James was achieved. It can be noticed that the two heat maps have a lot of similarities, having the default 'good' labeled tweets mostly annotated accurately, with 31 and 32 accurate from Stephan's and James' annotations respectively. However, most of the default labeled 'bad' and 'neutral' tweets were inaccurately annotated from both annotators, having 26 and 22 out of 97 accurate annotation for the 'bad' records. As for the figure 3.10, it represents the annotations accuracy between James and Stephan: the results were highly accurate with most of the data being correctly annotated while following a shared guideline.

This low IAA value could raise concerns about the quality of the dataset acquired. However when put into context that the data is acquired from Twitter, and taking into consideration the Twitter specifications, it becomes noticeable that many of the tweets only had links or one word linked to a media file that describes the emotion. These messages lack the sentimental clues, making it hard for annotators to agree on the emotion labeled for the text. For example, some tweets only contain ChatGPT, followed by a link which is interpreted as neutral due to the lack of sentiment. But, by checking the linked media, it could be indicating that *ChatGPT is bad*, therefore labeled as bad in the dataset and leading to confusions in the annotation process.

Despite the lower IAA, one of the hypotheses to be answered showcases the performances of the model, indicating that even with a low IAA, high performing models can still be built and can have a good prediction power on the same kind of data. Then, testing these models on real-life collected data helps identify the generalizability of the models as well as proving the importance of having the right dataset following the right guideline and having all the needed information.

Chapter 4

Process Followed

4.1 Vectorizers

Vectorizers are used to convert text data into numerical vectors, to be able to apply ML algorithms on text since machines are only able to understand values and not texts. (Diego, 2021)

4.1.1 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) in SA is one of the well-known techniques which converts textual data into numerical representations that ML models can interpret. It is effective in highlighting the importance of words in text data, making it an important tool in SA. TF measures how frequently a word appears in a tweet, while IDF measures the importance of this word by considering how often it appears in the data. The TF-IDF is a straight-forward vectorizer that highlights important terms in each document, which in turn improves the relevance of features used for modeling. However, this technique contains some drawbacks: it can sparse the data by treating similar term independently, it does not capture the context of the sentence, and finally it relies on a fixed vocabulary. The formula it follows is:

$$tfidf(t, d, D) = tf(t, d) * idf(t, D) \quad (4.1)$$

t denotes the terms; d denotes each document; and D denotes the collection of documents. (Nguyen, 2014)

4.1.2 Word Embedding

Word embeddings are a form of numerical word representation that enables words with similar meanings to possess similar vector representations. Their main goal is to map words or phrases from a vocabulary to dense numerical vectors, in order to capture relationships between words. Word embeddings are word representation vectors learned unsupervisedly, and the similarities between them are reflected (Mandelbaum and Shalev, 2016). The DNN model uses the trainable embedding, so the embeddings are learned while training. At first, the embedding layer starts by setting the word vectors to random values. Throughout the training process, these vectors are modified through backpropagation in order to reduce the loss function. After training, the embedding layer holds dense vectors that are similar for words with similar contexts. Using this method of vectorization helps capture semantic similarities by making sure that words that mean the same have vector representations that are similar while also decreasing the complexity of text data. Additionally, neural network learn better with dense representations, improving the overall learning process and model performance.

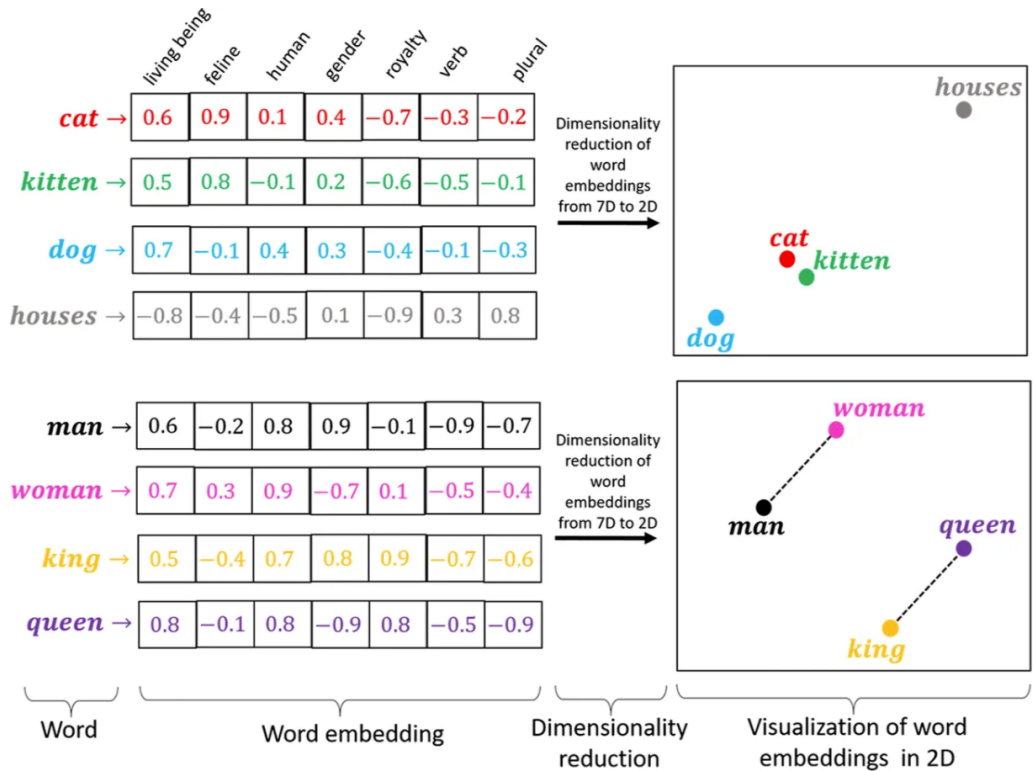


FIGURE 4.1: Word Embeddings

The figure 4.1 shows the word embeddings of different words and their relation to one another. The cat and kitten proved to be much closer in embedding values as shown in the 2D visualization, than between cat and houses. Additionally, man and woman, and king and queen, are shown to have a similar distance which indicates a matching related meaning (the difference is the sex). (Gautam, 2020)

4.2 Models

To perform sentiment classification, two ML models (Random Forest, Maximum Entropy) as well as one DL model (Deep Neural Network) were built to identify the best approach for this SA project. The data could be split as 80-20% or 70-30% for the training and testing respectively. This split is chosen according to how large the dataset is and how much the models need training without overfitting.

4.2.1 Random Forest

A Random Forest (RF) Classifier is an ensemble learning method that operates by constructing multiple decision trees during training and entering the mode of classes (classification) or mean prediction (regression) of the individual trees. It's an effective and versatile model, often used for classification and regression tasks due to its simplicity and robustness. SA with RF takes advantage of its algorithm's capabilities that enhance the accuracy and efficiency of sentiment classification, making it a promising approach for extracting meaningful emotional information from textual datasets. (Raphael Couronné, 2018)

RF train on a vectorized column where each row represents a text sample and each column represents a feature. The RF algorithm begins by creating multiple subsets

(bootstrap samples) of the training data. Each subset is created by randomly sampling with replacement from the original dataset. This means that some samples may appear multiple times in a subset, while others may not appear at all. After each bootstrap sample is created, a decision tree is made. Then, a subset of features at each node is selected (feature bagging). The best feature and threshold are selected to split the data at each node based on criterion such as Gini impurity or Entropy. This process is repeated recursively until the tree reaches a maximum depth, or other stopping criteria are met. Additionally, each Bootstrap sample will make one tree where each one will be slightly different since it is trained on different bootstrap sample and a different subset of features. For example: considering there are 4 texts with labels (0:Bad, 1:Good), "I love this" with label 1 (1), "I hate this" with label 0(2), "Great experience" labeled 1(3) and "This is the Worst" labeled 0(4). Bootstrap samples will be taken from the dataset to build trees such as sample 1 consists of texts 1, 2, and 3, sample 2 consists of texts 2, 3, and 4, etc. The first sample will select a random feature such as "hate" and "experience" where the algorithm will define the best feature of the split to be chosen. In this case, the word "hate" will be chosen as the first split since it contains the most sentimental meaning, and the process continues recursively until the stopping criteria is met such as reaching the set maximum tree depth. Then, the RF model will make final predictions based on the individual trees created by using majority voting. To be able to develop the model and not fall into the problem of overfitting, hyper-parameter tuning was made:

- **n_estimators:** the number of trees created (bootstrap samples), can take any positive integer with 10 to 500 most commonly used. This parameter was tuned on 50 and 100.
- **random_state:** A seed for the random number generator, ensuring that results are reproducible, can take any integer or None, with fixed number ensuring that the same random selections are made each time the model is run.
- **bootstrap:** A boolean value that decides whether each tree is trained on the entire dataset or a sample of it. If it is set to false, bootstrap will sample observations without replacement of records, meaning data records can't be used multiple times. Both values were tested but the model took the bootstrap with a false value.
- **max_features:** The number of features to consider when looking for the best split, can take any integer number, 'sqrt' meaning the square root of the number of features, 'log2', the base-2 logarithm of the number of features, 'None' considers all features. It was chosen to have a value of 10000.
- **min_samples_leaf:** The minimum number of samples required to be a leaf node, it can take any integer or float. It was tuned by taking the values 5 and 10.
- **min_samples_split** The minimum number of samples required to split an internal node, and can take any integer or float. This parameter was set to 5.

4.2.2 Maximum Entropy

The concept of entropy, or the measure of randomness, quantifies the unpredictability present in the data. Meaning that the higher the entropy is, the greater the uncertainty about the outcome of an event is, and more information is required to describe

the state of the system. For example, tossing a fair coin has an entropy of 1 bit where the probability of the two outcomes heads or tails, is 0.5. In this case, the only information needed is either yes or no, which will describe the result. The formula for entropy is:

$$E = \sum_{i=1}^N P_i \log_2 P_i \quad (4.2)$$

with P_i being the probability of randomly selecting an example in class I . (Robinson, 2008)

As for the principle of Maximum Entropy (MaxEnt), when dealing with a probability distribution based on partial information, the distribution chosen would be the one with the highest entropy that best represents the current state of knowledge about specific system. Additionally, it does not make any additional assumptions or biases, which proves to be beneficial when dealing with words that have complex meanings (Paul Penfield, 2003). When handling text, the MaxEnt uses the constraints set from the training data to determine the best probability distribution that follows them (the one with the highest entropy).

Hypothetically, the training data show that when "like" is in a tweet, 70% of the time it is associated to a good sentiment (the constraint is set). As for the residual 30%, it is evenly split between the other 2 classes which ensures that the model remains unbiased and flexible. Then, without taking into consideration the context, if the MaxEnt model faces the word "like", it will choose the unbiased probability distribution with the given constraint: 70% for good, and 15% for bad and neutral. In other words, the principle of MaxEnt is a method of inference that works on making logical assumptions on the basis of partial information.

The creation of the MaxEnt model begins with vectorizing the words in the tweets. This step entails the calculation of the Term Frequency-Inverse Document Frequency (TF-IDF) of each word to transform the text into a numerical format that is suited for Machine Learning (ML). The next step was the MaxEnt Classifier which used the "LogisticRegression" classifier. The latter is suited for categorical predictions based on maximizing the entropy of the probability predictions across classes.

Furthermore, a few of hyper-parameters were used:

- **max_features:** It belongs to the 'TfidfVectorizer' and determines the unique features or words to be considered in the TF-IDF process. Due to the large nature of the dataset, it could be set anywhere between 5000 to 20000. The value 10000 was chosen for this model.
- **max_iter:** It is present in the Logistic Regression classifier. This hyper-parameter specifies the number of iterations until the convergence to an optimized solution while minimizing the loss. The choice of this parameter depends on factors such as the size of the dataset or the complexity of the model and can be set in the range from 1 to 1000. In this project, the tuning used the values 5, 10, and 20.
- **solver:** It uses optimization algorithm that help find the best parameters in a prediction of probabilities. Various solvers can be used for many purposes: 'lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', and 'saga'. This hyperparameter was tuned on 'saga', 'sag', and 'lbfgs'.

4.2.3 Deep Neural Network

An artificial neural network is a traditional network that solves easy and straightforward tasks using a collection of layers. However, a Deep Neural Network (DNN) is a type of artificial neural network inspired by the human brain that has multiple layers with more depth between the input and output. It is used for much more complex tasks and consists of interconnected nodes, or neurons, and edges. The DNN has 3 types of layers: the input layer, the hidden layer, and the output layer. (K, 2023)

In this project, the DNN will predict the sentiment of the text by calculating the probability that it belongs to the classes. First, the input layer will take the input data, or the tweets, then use word embeddings to transform them into numerical vectors which will be fed to the layer. Word embeddings were used because they have a much lower dimensionality than one-hot encoded vectors resulting in less computational resources to store and manipulate. The second part is the hidden layers: each layer contains neurons that take the input from the previous layer and then perform mathematical operations so that it can learn the hidden patterns and create relationships in the data. During the training process, the network will learn from the labelled records and use its backpropagation process using the optimization algorithms to adjust its parameters and reduce the prediction errors. The DNN uses epochs, which is a pass through the entire dataset where samples are used once to update its parameters. The final layer is the output which is the classification or the final prediction of the model. It consists of one neuron per sentiment class, with a total of 3 neurons in this project: good, bad, and neutral. After the training is complete, the relationships created will be used as equations to predict the output variables based on the input. This model is very effective due to its automatic handling of the complex nature of tweets and its ability to understand patterns dependent on the context and combination of words.

The DNN offers a validation set where a part of the training data is split to monitor the performance of the model and determine if there is any overfitting. This is tracked using the validation loss: the 'EarlyStopping' function stops the fitting of the model if this parameter increases over a specific number of epochs, and the 'ReduceLROnPlateau' which reduces the learning rate when a metric isn't improving anymore which may decrease the loss and gives the network a chance to adjust. These functions will not only work on enhancing the DNN, but they will avoid any overfitting that could occur during training.

As mentioned before, the first layer of this model is the word embedding. The next layer is the Flatten layer which transforms the 2D output of the embedding layer into a 1D tensor, making it easier to input into following layers without changing the batch size. After that, a dense layer containing 128 units utilizes Rectified Linear Unit (ReLU) activation to add non-linearity which is important for understanding intricate patterns. The dropout layers set at 0.5, help prevent overfitting by randomly disabling some neurons while training. Then, a dense layer consisting of 64 units continues to operate on the data, ensuring non-linearity through ReLU activation. Following it is an additional dropout layer. Finally, the last layer of the model consists of 3 units and uses Softmax activation instead of the Sigmoid (used for binary classification) to create a distribution of probabilities among different classes.

Various hyper-parameters were used and tuned to increase the model's performance:

- **max_words:** It's in the embedding function to determine the maximum number of unique words to be used, it could be ranging from 5000 to 20000. It was set to 10000.

- **embedding_dim:** The embedding dimensions parameter sets the dimensionality, or size, of the dense vectors of the words. It influences the performance of the model as well as its complexity. For a large dataset and a balanced computational power, it could be set in the range of 100 to 300.
- **batch_size:** This hyper-parameter defines the number of samples processed before the model updates its parameters which affects the convergence and generalization. It is usually set as 32, 64, or 128 depending on the speed needed for training as well as the learning of the model.
- **epochs:** As previously explained, the DNN uses this hyper-parameter to pass through the training dataset. Changing the values between 5, 10, or 20, will impact if the model is overfitting or underfitting, as well as the duration of training.
- **learning_rate:** A very important parameter in this model is the learning rate of the Adam optimizer. By reducing the value of this rate, the model will have more time to learn deeper and complex patterns of the data while avoiding overfitting. This optimizer's values typically are 0.001, 0.0001, or 0.00005 for extremely large datasets, or any values in between.
- **Dropout:** This hyper-parameter accepts as a float value between 0 and 1. It is a regularization technique that is used to prevent the overfitting since it sets a fraction of input units in training to 0.

4.2.4 Large Language Model

The Large Language Models (LLMs) are advanced DL models that can provide a deeper understanding of human language more efficiently. OpenAI's GPT-4 model is a prime example of LLM: it is trained on a great amount of textual data from the Internet, allowing it to capture patterns and context of the language. This learning process is advantageous for scientists since it enables a variety of NLP tasks such as text generation, completion, and classification, more specifically SA. It also has the multilingual ability that allows it to learn and understand various human languages and perform tasks such as translation. (Kerner, 2024)

However, LLMs come with many challenges and disadvantages. A major one is that their computational cost, both in training and use, places a substantial toll on the machine running it by exhausting its computational resources. Additionally, this type of models may prove to be incorrect, and they will produce a false output with confidence, which will only confuse the users. For example, ChatGPT was found to have multiple incorrect statements because it acquired its data from the public Internet, which may includes false information. Another disadvantage found in LLMs is their unintentional bias. As mentioned before, their training data is from the Internet, which is biased by nature, which will lead to biases in the output content. For instance, a text completion model may finish the sentence *The CEO fired with his employees*, proves to be biased since it associates the role of CEO showing the gender stereotypes. (Kok, n.d.)

This project acquired the LLMs from HuggingFace: a platform for open-source AI where users build, use and train models in various areas. Two LLMs were used: first, the zero shot classification, which classifies unseen data even during training, relying on its generalization ability. The text classification model classifies text into previously defined classes or labels, in this SA context, categorizes tweets into 'good',

'bad, or 'neutral'. This use of pre-trained LLMs will help test the hypothesis set that these models are not always generalizable and need to be fine-tuned.

4.3 Metrics

It is crucial to assess the ML model's performance by utilizing quantitative measures, also known as metrics. They enable models to be optimized more effectively by pinpointing areas where there is need for improvement. Provided with numerical performance values, a more precise comparison can be made to evaluate which model is better suited for the given problem. Various metrics such as Accuracy, Precision, Recall, and F1-score, provide different perspectives on the model's performance, allowing a deeper understanding of its robustness. These values were calculated on both the train and test sets to validate if the model is overfitting or not. (Bonnet, 2023)

- **Accuracy:** Defined as the ratio of correctly predicted instances to the total instances. It is a useful metric when the classes are well-balanced - each class approximately has the same number of instances. However, accuracy can be misleading when classes are imbalanced.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.3)$$

- **Precision:** Known as positive predictive value, that measures the accuracy of the positive predictions. It is defined as the ratio of true positive predictions to the total number of positive predictions. Precision is particularly useful when the cost of false positives is high.

$$Precision = \frac{TP}{TP + FP} \quad (4.4)$$

- **Recall:** Known as sensitivity or true positive rate. This metric measures the ability of the model to identify all relevant instances. It is defined as the ratio of true positive predictions to the total number of actual positives. Recall is crucial when the cost of false negatives is high.

$$Recall = \frac{TP}{TP + FN} \quad (4.5)$$

- **F1-Score:** It is the harmonic mean of precision and recall, providing a single metric that balances both concerns. F1-Score is especially useful when the classes are imbalanced, as it considers both false positives and false negatives.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4.6)$$

4.4 Google Form

To be able to validate the models, a Google Form survey was sent to gather tweet-like inputs from different people. Furthermore, this collection of a variety of text data can be used to test and evaluate the generalizability of the ML models. The survey requested from people to share their sentiment towards ChatGPT, and then express it in the form of a tweet written in English that shows their point of view.

The list of different tweets that have been obtained, a small analysis and the format of the google form are mentioned in Appendix C
Some examples of the gathered tweet-like inputs are:

- *i love it #chatgpt #bird* - [Good]
- *Chatgpt is ruining our lives. It will leave soon enough leave us with no jobs #AITakeover #Chatgpt #annoying* - [Bad]
- *ChatGpt is not only a bad technology, it is very mind numbing... everyone is using it as an easy way out which is very annoying* - [Bad]
- *Good in helping us but it is not really reliable cz it can make mistakes* - [Neutral]
- *I'll begin with one word: #helpful. Then, #impressive because this is one of the most powerful engine of its kind. But there is also a little bit of fear as it continue to developp other features. The creators themselves can't predict how far it could go. #SafetyPlizz* - [Good]

Chapter 5

Results and Discussion

The results of this SA study give better insights into how effective the models created were, as well as the various hyper-parameters and techniques used that affected their performances. The effects of the hyper-parameter tuning allow for an identification of the strengths and weaknesses of the different classification models. Furthermore, this analysis will highlight the best model for SA.

As shown in the EDA, the dataset is imbalanced: the 'bad' label has records equal to both 'good' and 'neutral' records combined. For all three models, this led to low performances, especially for the 'neutral' label, because they had a harder time identifying the nuances from the small 'neutral' records available. Therefore, the oversampling technique was done using the 'RandomOverSampler' function. This method randomly chooses train records from the minority classes: 'good' and 'neutral', and duplicates them until they have the same amount of train records as the majority class: 'bad'. This oversampling results in a more balanced dataset, as well as a model that will learn better from the minority classes and less biased towards the majority class, which will in turn lead to better performances.

After hyper-parameter tuning, the best hyper-parameters for each model emerged. The 'train test split' method was applied the same for all models. This function randomly divides the whole dataset into train and test sets without any replacements. For instance, when set at 0.3 (test size), it returns a test set of 30% of the data, and a train set of 70%. In this project, the data was split with values of 0.2 and 0.3, or 80%/20% and 70%/30% for the train/test sets. However, after tuning, the value 0.2 was chosen because it led to better performances of models. This splitting function also uses the 'random state' parameter which sets a seed to the random generator so that the train and test splits always have the same results. If this parameter was not set, then each execution will have a different random split of the dataset and lead to different results in the models. The seed could have any integer value, and in this project, the value 42 was used because it is the most commonly chosen.

For the RF, the 'n estimators' (number of trees), after testing, was given a value of 100 which had the best results while balancing the model in terms of accuracy and speed. Additionally, the 'minimum samples split' did not have an effect on the performance which is why it was set at 5. The 'bootstrap' was set at False to avoid using the same train sample twice. The tweets were found to have 8944 unique words: 'maximum features' was set at 10000 to avoid any unnecessary features. The most important hyper-parameter in RF is the 'minimum samples leaf'. When set at 1, the model was overfitting and train data had an accuracy of 100%. The train accuracy is a measure used after the model building stage by predicting the target label of the train set. This step will ensure that no overfitting occurred when the model was learning the train data. The value 10 for the 'minimum samples leaf' was chosen because it balances out the performance between train and test and reduces the overfitting. After tuning, the RF had the lowest accuracy of 76.26% out of the three models.

The MaxEnt model had three hyper-parameters that needed tuning: the 'maximum features' which was previously explained and the value 10000 was chosen for the same reasons. The 'maximum iterations', after testing different values, was chosen as 10 since this dataset was converging at low values. And finally, the 'solver' hyper-parameter was set to 'saga' because it is faster in speed and supports multi-class classifications while also providing regularizations and penalties. By setting these hyper-parameters to their best observed values, MaxEnt resulted in an accuracy of 82.61%, becoming the second best model.

As for the DL model, many hyper-parameters were tuned to achieve optimal results. The 'maximum words', same as 'maximum features', was set to 10000. The 'embedding dimension' hyper-parameter had little effect on the performance but a higher toll on the speed of the model, which is why it was chosen to have its optimal value of 100. Furthermore, the 'Dropout' parameter did not have a high effect on the accuracy which is why it was set at its balanced value of 0.5. As for the 'batch size', after testing, the model had the best balanced results at the value of 32. The 'epochs' parameter had a high effect on the overfitting of the model: the higher it was, the more learning and overfitting it had. After testing, the best train and test results were found to be at 5 epochs. The model was found to be learning and overfitting really fast, which is why the 'learning rate' of the Adam optimizer was tuned. It was set at 0.00005 to help the model have more time to learn the patterns of the data. This combination of hyper-parameters was used because of the overfitting ability of the DNN model, resulting in it being best model for SA with an accuracy of 85.02%.

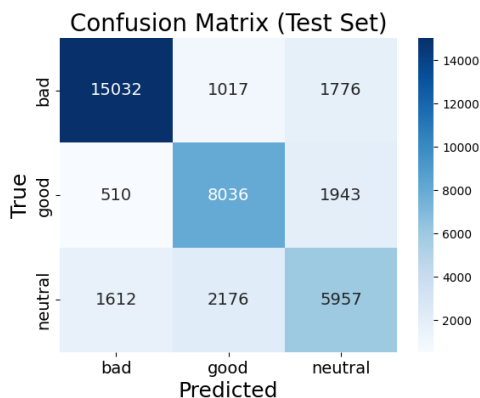


FIGURE 5.1: RF Confusion Matrix

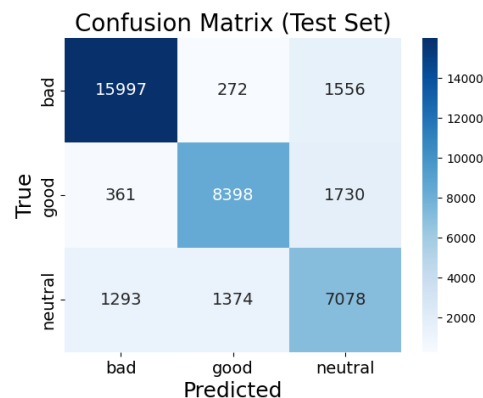


FIGURE 5.2: MaxEnt Confusion Matrix

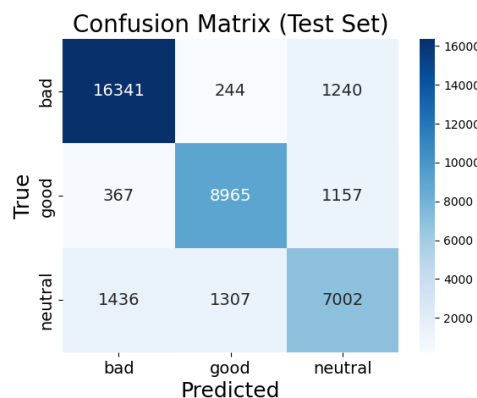


FIGURE 5.3: DNN Confusion Matrix

Opposed to the ‘neutral’ performances, all the models excelled at predicting ‘bad’ labels. Even though they had better ‘neutral’ label results, it still proved to have lower performances when compared to the other two labels as shown in the confusion matrices in the above figures 5.1, 5.2, and 5.3. This could be due to the models’ low predicting ability where sentimental nuances are lacking, contrasted to having more emotional clues regarding bad tweets (‘ChatGPT is annoying’ is immediately regarded as a bad tweet, but ‘ChatGPT has new updates’ may be regarded as a good tweet since users are staying up to date with this technology). To test this hypothesis, a test on all models was done by removing the ‘neutral’ tweets from the dataset. This turned a multi-class classification into a binary classification which would make it easier for models to classify opposite polarities. This hypothesis was proven when all the models were shown to have an improved test accuracy: the RF increased to 89.51%, the MaxEnt to 94.97%, and the DNN to 94.54%. A full comparison between multi-class classification and binary classification results is shown in the table 5.1.

TABLE 5.1: Multi-Class and Binary Classifications Comparison

Model	Classification	Accuracy	Precision	Recall
RF	Binary	89.51%	90.26%	89.51%
	Multi-Class	76.26%	76.53%	76.26%
MaxEnt	Binary	94.97%	94.97%	94.97%
	Multi-Class	82.61%	82.75%	82.61%
DNN	Binary	94.54%	94.53%	94.54%
	Multi-Class	85.02%	84.90%	85.02%
Zero Shot	Binary	60.60%	64.57%	60.60%
	Multi-Class	41.03%	43.13%	41.03%
Text Class	Binary	40.88%	77.76%	40.88%
	Multi-Class	44.87%	47.88%	47.87%

The DNN model had the best performances over the MaxEnt and RF models in this SA project, especially having a multi-class classification with imbalanced classes. This performance is a result of multiple features of the deep learning model. First, it excels on the dataset at hand, where tweets are being analyzed to find complex patterns and relationships to classify sentiment. Additionally, this model captures the subtle emotional clues in languages that simpler ML models like RF might oversee. Furthermore, DNNs can understand the context over long sentences whereas the MaxEnt considers the input features as independent of each other. Lastly, this model is the best performing because it is flexible and has the ability to fine-tune while the training is running (for example: one of the callbacks that could be implemented is the ‘ReduceLROnPlateau’, where it reduces and slows down the learning rate of the model if it notices any overfitting).

After the model building step, the text classification LLM was implemented to make a comparison between the performance of a pre-trained model, with ML models trained on the data at hand. However, the performance of the model was inefficient: it had a 44.87% accuracy on the 30000 records sampled from dataset present, which highlighted the need for fine tuning. Furthermore, the fine tuning was shown to need a great computational power that is not available at hand for this project. Additionally, the zero-shot classification LLM had an accuracy of 41.73% on 3000 records. Both of these advanced models did not prove to have decent results in this SA project, as well as not having fast runtime (the zero-shot classification took around 2 hours to classify only a part of the data without fine tuning). As a result, the use of LLMs in the context of SA was not justified neither by the resource consumption, nor by the complexity and runtime, as compared to the other 3 ML, more

efficient, models. The figure 5.4 compares the different models' train and test accuracies. Additionally, a full comparison of performances between these different ML algorithms is shown in the table 5.3.

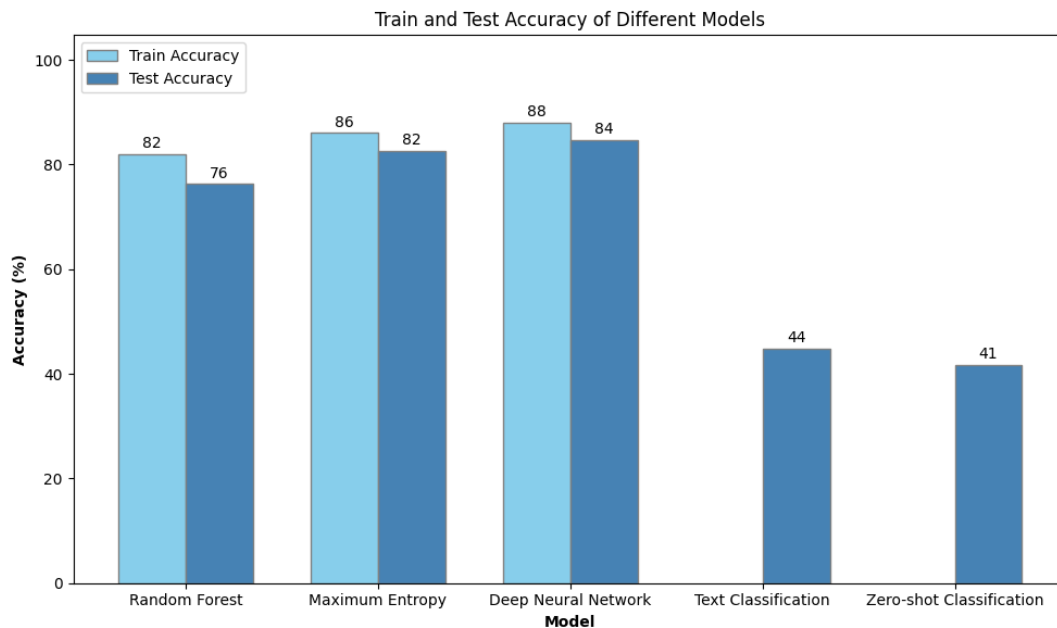


FIGURE 5.4: Comparison of Multi-Class Models

The final part of this project is applying the models created on the collected tweets from the google form. This data contained only 26 records and was very imbalanced towards the positive tweets. The models, even though they had high performances, did not generalize well. The results of these ML algorithms are shown in the table 5.2. The RF model had a 42.31% accuracy on this form dataset, and the MaxEnt model a 53.85%. As for the DNN DL model, it achieved an accuracy of 30.77%.

As discussed before, one of the main goals of this project is to validate the generalizability of models on unseen data. As mentioned in section 3.5, the agreement between each annotator and the sampled data is low. One of the hypotheses presented is that a dataset with low IAA may present a high performing model on the aforementioned dataset, but will not perform efficiently on other data, even if related to the subject at hand. The models built on this 'tweets' dataset showed high performance as seen in table 5.3. Testing these models on the data collected from the Google Form (Appendix C), a decrease in performance was mainly observed. As shown in table 5.2, the model with the best performance achieved a 45.85% F1-Score. This low performance might be due to different reasons.

- **Language variance:** The tweets collected for this work, were mainly from American Twitterers, while the ones collected in this form consists of Lebanese Twitterers. This language variance can disturb the model, which lead to a low performance.
- **IAA disturbance:** Good annotation is crucial for any classification task. Machines learn the mistakes taught to them, so differences in annotations can lead to a model's disturbance, and a bad performance. In this project, it has been presented that even if low IAA is observed, a good model can be obtained. But trying the model on unseen data showed that the models were not able

to perform, indicating that they are not generalizable, confirming the second hypothesis so far.

- Validation Set restrictions: The google form collected a low number of records that was skewed towards the positive sentiment (57.7% as observed in figure C.1). This lack of records and imbalance may have impacted the reliability of the models trained and their generalizability on unseen data. Therefore, obtaining additional unseen tweets might change the models' performances, which could potentially invalidate the hypothesis confirmation mentioned earlier.

TABLE 5.2: Predictions on Google Form

Model	Accuracy	Precision	Recall	F1-Score
RF	42.31%	42.72%	42.31%	40.61%
MaxEnt	53.85%	40.28%	53.85%	45.85%
DNN	30.77%	47.70%	30.77%	34.92%

TABLE 5.3: Multi-Class Models Performance Comparison

Model	Vectorizer	Hyper-Parameters	Accuracy	Precision	Recall
RF	TF-IDF	n_estimators=100, bootstrap=False, min_samples_leaf=10	76.26%	76.53%	76.26%
		n_estimators=100, bootstrap=True, min_samples_leaf=10	75.50%	75.64%	75.50%
		n_estimators=50, bootstrap=False, min_samples_leaf=10	76.22%	76.48%	76.22%
		n_estimators=100, bootstrap=False, min_samples_leaf=5	77.03%	77.18%	77.03%
MaxEnt	TF-IDF	max_iter=10, solver='saga'/'sag'	82.61%	82.75%	82.61%
		max_iter=5, solver='saga'/'sag'	82.46%	83.10%	82.46%
		max_iter=20, solver='saga'	82.67%	82.90%	82.67%
		max_iter=10, solver='lbfgs'	63.07%	67.45%	63.07%
DNN	Word2Vec	learning_rate=0.00005, epochs=5, batch_size=32	85.02%	84.90%	85.02%
		learning_rate=0.00005, epochs=5, batch_size=64	84.38%	84.18%	84.38%
		learning_rate=0.00005, epochs=10, batch_size=64	85.05%	85.03%	85.05%
		learning_rate=0.0001, epochs=5, batch_size=64	84.60%	84.53%	84.60%
Zero Shot	NA	NA	41.73%	44.35%	41.73%
Text Class	NA	NA	44.87%	47.88%	47.87%

Chapter 6

Conclusion & Future Studies

6.1 Conclusion

The purpose of this project was building ML models that can be used for analyzing the sentiment of people. It highlighted the models' performances and helped test the hypotheses presented regarding the IAA and the LLMs. Two ML models (RF and MaxEnt) as well as one DL model (DNN) were trained and tested. The latter model emerged as the best, achieving 85.02% accuracy, while Maximum Entropy achieved an accuracy of 82.61%, with the second highest accuracy. Finally the random forest had the lowest accuracy scoring, predicting 76.26% of the data correctly. These results helped make three conclusions to our hypotheses.

Firstly, despite the low IAA of 0.35 in the annotated dataset, the models created performed well on a multi-class classification with accuracies of 76%-85%. This validates the hypothesis that high IAA is not always a needed for creating good models. Secondly, it was determined that models achieving high performances on specific datasets failed to generalize well to unseen data as seen previously in Table 5.2. It showed an accuracy between 30% and 53% for the models on the google form data. This finding proved that a model's performance on one dataset does not necessarily translate to similar success on another, underscoring the need for careful consideration when deploying models in different contexts. Moreover, it highlighted the need for an IAA in all ML projects, as opposed to the lack of this step in the different research papers shown before. These papers achieved good performing models without calculating the IAA or showing the generalizability of their models. While the ones in this project, even though they had high performances, having a dataset with a low IAA did not allow them to be generalizable. Thus, there is a need for a good, well determined annotation guideline, as well as a data description for any labelled dataset to create good, generalizable models.

Lastly, pre-trained LLMs showed low initial performances and required fine-tuning for the training data to be generalizable. However, this was constrained by computational resources which underscores the need for significant computational power to fully leverage this type of models and achieve optimal performance in SA tasks. These findings are crucial for the field of SA and NLP. But, they also point to the need for caution when deploying models across different datasets, as performances can vary widely. Additionally, even though the models developed performed well, this study showed important steps that lack in previous studies. For instance, very few papers revealed an IAA, or gave a model's performance on unseen data, which made their study questionable at best. However, through this study, the importance of having a high IAA was shown to be necessary to generalize the model across various dataset. Furthermore, LLMs aren't always accurate as shown in Table 5.3, with them predicting only 41% and 44% of the data correctly. It highlighted the need for tuning which is resource-intensive and not available for this project.

6.2 Future Studies

Lots of research can be done in the SA field, and many flaws were found in the models which can be improved. Therefore, future research should focus on addressing the limitations explained in section 6.1.

As seen before, lots of tweets had links with content that couldn't be easily accessed or seen with the time and resources limitations in this project. In addition, these tweets were meaningless without knowing the link's content, which showed the need for improvements. In future studies, instead of removing these links altogether, better methods can be used to handle and extract data from them. For example, the implementation of web scraping tools to get the links' content and then using ChatGPT's Application Programming Interface (API) to summarize it, or image recognition techniques to extract text from images and analyze the content in it.

Subsequently, sarcasm detection also posed a challenge for the models used due to their inability to understand it through context clues. Therefore, developing more advanced pre-processing techniques to identify and handle sarcastic tweets is essential. This is done by utilizing more DL models, such as transformer-based models like BERT or GPT that can better understand the context. Additionally, there is a need to explore new areas within SA, such as the impact of different pre-processing techniques, which may advance this field.

Another important notion for future researches is keeping track of technological advancements in NLP and the improvements of computational power. As the field is getting better and loads of researches are done daily, staying up to date with new technologies is necessary. Namely: regularly monitoring and incorporating the latest advancements in NLP techniques (newer versions of transformer models improved training methodologies), exploring scalable solutions that leverage cloud computing and parallel processing to handle large datasets efficiently. And lastly, engaging in collaboration with other experts in the field can be greatly beneficial to exchange knowledge with experienced people in the field.

Additionally, future studies can work on building new classification models using different algorithms. Some of these could include: SVM, NB, Bi-lstm RNN, etc. By applying them, a deeper comparison between the different models could be made and the optimal one for SA can be found.

Moreover, applying the models on unseen data is important to check for generalizability. This project included this method using a google form with a very small number of imbalanced records which may have impacted the generalizability. Therefore, future work could address this issue by gathering a larger and more balanced dataset to determine whether the results observed were due to the small and skewed dataset, or if it is affected by the low IAA.

The final significant issue in this project was the low IAA and good performing models. They did not perform well on unseen data. To address this, future studies should focus on finding more generalizable datasets where higher IAA could be achieved. This approach will improve the understanding of the relationship between the IAA and the generalizability of the model.

In conclusion, while this study provided valuable insights, there is still a lot to learn in this rapidly advancing field. By improving the handling of linked content, enhancing sarcasm detection, focusing on IAA calculations, and keeping track of technological advancements, future studies can significantly advance the SA field.

Appendix A

IAA guideline

Good:

- The comment mentions how the features of ChatGPT help perform good tasks.
- The comment uses good descriptive words about ChatGPT and its results. (good, incredible, holy shit, insane...) or double negative (not bad, not the worst...)
- The comment mentions the potential of ChatGPT and AI.

Bad:

- The comment uses bad descriptive words about ChatGPT and its wrong results (bad, wrong, problematic, annoying...) or negation of positive (not interesting, not good...).
- The comment mentions the consequences of ChatGPT (loss of jobs, uncredited work...)

Neutral:

- The comment leads to another article/image.
- The comment poses an opinion question.
- The comment is an informative/declarative sentence.

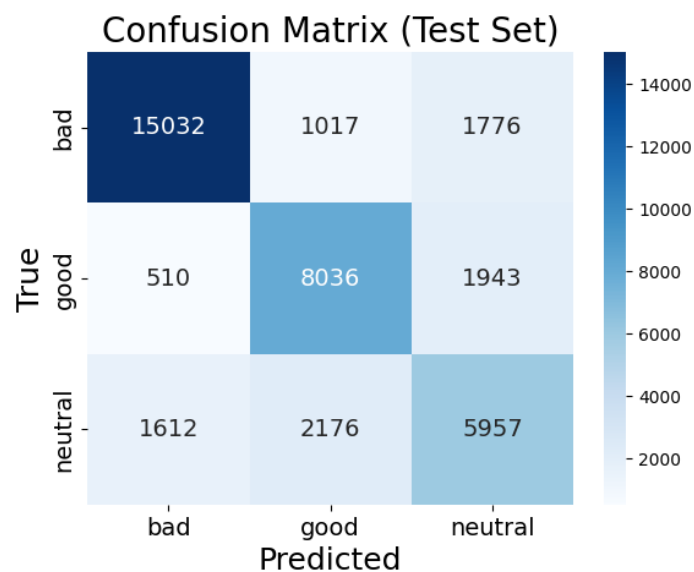
Appendix B

Hyper-parameter Tuning Results

Random Forest:

1. *n_estimators*=100, *bootstrap*=False, *min_samples_leaf*=10:

Train Set Metrics:				
Accuracy: 82.82				
Precision: 82.82				
Recall: 82.82				
F1 Score: 82.82				
Classification Report:				
	precision	recall	f1-score	support
bad	0.87	0.86	0.87	71657
good	0.82	0.84	0.83	71657
neutral	0.79	0.78	0.78	71657
accuracy			0.83	214971
macro avg	0.83	0.83	0.83	214971
weighted avg	0.83	0.83	0.83	214971
Test Set Metrics:				
Accuracy: 76.26				
Precision: 76.53				
Recall: 76.26				
F1 Score: 76.36				
Classification Report:				
	precision	recall	f1-score	support
bad	0.88	0.84	0.86	17825
good	0.72	0.77	0.74	10489
neutral	0.62	0.61	0.61	9745
accuracy			0.76	38059
macro avg	0.74	0.74	0.74	38059
weighted avg	0.77	0.76	0.76	38059



2. $n_estimators=100$, $bootstrap=True$, $min_samples_leaf=10$:

Train Set Metrics:					
Accuracy: 79.44					
Precision: 79.33					
Recall: 79.44					
F1 Score: 79.35					
Classification Report:					
	precision	recall	f1-score	support	
bad	0.83	0.86	0.85	71657	
good	0.79	0.81	0.80	71657	
neutral	0.76	0.71	0.73	71657	
accuracy			0.79	214971	
macro avg	0.79	0.79	0.79	214971	
weighted avg	0.79	0.79	0.79	214971	
Test Set Metrics:					
Accuracy: 75.50					
Precision: 75.64					
Recall: 75.50					
F1 Score: 75.54					
Classification Report:					
	precision	recall	f1-score	support	
bad	0.86	0.84	0.85	17825	
good	0.71	0.76	0.73	10489	
neutral	0.61	0.59	0.60	9745	
accuracy			0.76	38059	
macro avg	0.73	0.73	0.73	38059	
weighted avg	0.76	0.76	0.76	38059	

3. $n_estimators=50$, $bootstrap=False$, $min_samples_leaf=10$:

Train Set Metrics:					
Accuracy: 82.57					
Precision: 82.56					
Recall: 82.57					
F1 Score: 82.56					
Classification Report:					
	precision	recall	f1-score	support	
bad	0.87	0.86	0.87	71657	
good	0.82	0.84	0.83	71657	
neutral	0.79	0.78	0.78	71657	
accuracy			0.83	214971	
macro avg	0.83	0.83	0.83	214971	
weighted avg	0.83	0.83	0.83	214971	
Test Set Metrics:					
Accuracy: 76.22					
Precision: 76.48					
Recall: 76.22					
F1 Score: 76.32					
Classification Report:					
	precision	recall	f1-score	support	
bad	0.87	0.84	0.86	17825	
good	0.72	0.76	0.74	10489	
neutral	0.61	0.62	0.62	9745	
accuracy			0.76	38059	
macro avg	0.74	0.74	0.74	38059	
weighted avg	0.76	0.76	0.76	38059	

4. $n_estimators=100$, $bootstrap=False$, $min_samples_leaf=5$:

Train Set Metrics:				
Accuracy: 88.53				
Precision: 88.63				
Recall: 88.53				
F1 Score: 88.55				
Classification Report:				
	precision	recall	f1-score	support
bad	0.92	0.88	0.90	71657
good	0.88	0.89	0.89	71657
neutral	0.85	0.88	0.86	71657
accuracy			0.89	214971
macro avg	0.89	0.89	0.89	214971
weighted avg	0.89	0.89	0.89	214971
Test Set Metrics:				
Accuracy: 77.03				
Precision: 77.18				
Recall: 77.03				
F1 Score: 77.08				
Classification Report:				
	precision	recall	f1-score	support
bad	0.88	0.85	0.87	17825
good	0.73	0.77	0.75	10489
neutral	0.63	0.62	0.62	9745
accuracy			0.77	38059
macro avg	0.74	0.75	0.75	38059
weighted avg	0.77	0.77	0.77	38059

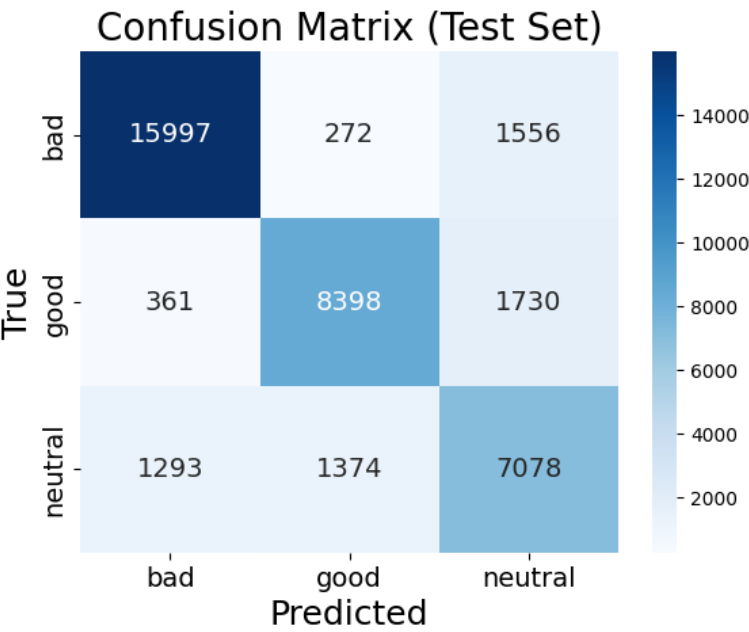
5. $n_estimators=100$, $bootstrap=False$, $min_samples_leaf=1$ (Overfitting):

Train Set Metrics:				
Accuracy: 99.87				
Precision: 99.87				
Recall: 99.87				
F1 Score: 99.87				
Classification Report:				
	precision	recall	f1-score	support
bad	1.00	1.00	1.00	71657
good	1.00	1.00	1.00	71657
neutral	1.00	1.00	1.00	71657
accuracy			1.00	214971
macro avg	1.00	1.00	1.00	214971
weighted avg	1.00	1.00	1.00	214971
Test Set Metrics:				
Accuracy: 77.81				
Precision: 77.40				
Recall: 77.81				
F1 Score: 77.48				
Classification Report:				
	precision	recall	f1-score	support
bad	0.86	0.88	0.87	17825
good	0.73	0.79	0.76	10489
neutral	0.67	0.58	0.62	9745
accuracy			0.78	38059
macro avg	0.75	0.75	0.75	38059
weighted avg	0.77	0.78	0.77	38059

Maximum Entropy:

- 1. `max_iter=10, solver='saga'/'sag':`

Train Accuracy: 86.88%				
Train F1 Score: 86.95				
Train Precision: 87.03				
Train Recall: 86.88				
Classification Report for Train Data:				
	precision	recall	f1-score	support
bad	0.93	0.92	0.92	71657
good	0.87	0.86	0.86	41695
neutral	0.76	0.79	0.78	38882
accuracy			0.87	152234
macro avg	0.85	0.86	0.85	152234
weighted avg	0.87	0.87	0.87	152234
Test Accuracy: 82.61%				
Test F1 Score: 82.68				
Test Precision: 82.75				
Test Recall: 82.61				
Classification Report for Test Data:				
	precision	recall	f1-score	support
bad	0.90	0.90	0.90	17825
good	0.83	0.81	0.82	10489
neutral	0.69	0.71	0.70	9745
accuracy			0.83	38059
macro avg	0.81	0.81	0.81	38059
weighted avg	0.83	0.83	0.83	38059



2. *max_iter=5, solver='saga'*:

Train Accuracy: 86.67%				
Train F1 Score: 86.74				
Train Precision: 86.87				
Train Recall: 86.67				
Classification Report for Train Data:				
	precision	recall	f1-score	support
bad	0.92	0.92	0.92	71657
good	0.88	0.84	0.86	41695
neutral	0.75	0.80	0.77	38882
accuracy			0.87	152234
macro avg	0.85	0.85	0.85	152234
weighted avg	0.87	0.87	0.87	152234
Test Accuracy: 82.63%				
Test F1 Score: 82.70				
Test Precision: 82.84				
Test Recall: 82.63				
Classification Report for Test Data:				
	precision	recall	f1-score	support
bad	0.90	0.90	0.90	17825
good	0.84	0.79	0.82	10489
neutral	0.68	0.72	0.70	9745
accuracy			0.83	38059
macro avg	0.81	0.81	0.81	38059
weighted avg	0.83	0.83	0.83	38059

3. *max_iter=20, solver='saga'*:

Train Accuracy: 86.85%				
Train F1 Score: 86.93				
Train Precision: 87.06				
Train Recall: 86.85				
Classification Report for Train Data:				
	precision	recall	f1-score	support
bad	0.93	0.92	0.92	71657
good	0.87	0.85	0.86	41695
neutral	0.76	0.80	0.78	38882
accuracy			0.87	152234
macro avg	0.85	0.86	0.85	152234
weighted avg	0.87	0.87	0.87	152234
Test Accuracy: 82.68%				
Test F1 Score: 82.77				
Test Precision: 82.91				
Test Recall: 82.68				
Classification Report for Test Data:				
	precision	recall	f1-score	support
bad	0.90	0.90	0.90	17825
good	0.83	0.80	0.82	10489
neutral	0.68	0.72	0.70	9745
accuracy			0.83	38059
macro avg	0.81	0.81	0.81	38059
weighted avg	0.83	0.83	0.83	38059

4. *max_iter=10, solver='lbfgs'*:

```

Train Accuracy: 63.75%
Train F1 Score: 64.92
Train Precision: 68.10
Train Recall: 63.75
Classification Report for Train Data:
      precision    recall  f1-score   support

    bad         0.80      0.68      0.74      71657
    good         0.72      0.56      0.63      41695
   neutral         0.42      0.63      0.50      38882

   accuracy          0.64      152234
  macro avg         0.65      0.63      0.62      152234
 weighted avg         0.68      0.64      0.65      152234

Test Accuracy: 63.07%
Test F1 Score: 64.27
Test Precision: 67.45
Test Recall: 63.07
Classification Report for Test Data:
      precision    recall  f1-score   support

    bad         0.80      0.68      0.74      17825
    good         0.71      0.56      0.62      10489
   neutral         0.41      0.61      0.49       9745

   accuracy          0.63      38059
  macro avg         0.64      0.62      0.62      38059
 weighted avg         0.67      0.63      0.64      38059

```

Deep Neural Network:

1. *learning_rate=0.00005, epochs=5, batch_size=32*:

```

Train Accuracy: 88.96%
4758/4758 [=====] - 15s 3ms/step
Train F1 Score: 88.93
Train Precision: 88.90
Train Recall: 88.96
Classification Report for Train Data:
      precision    recall  f1-score   support

    bad         0.93      0.94      0.94      71657
    good         0.89      0.89      0.89      41695
   neutral         0.81      0.80      0.80      38882

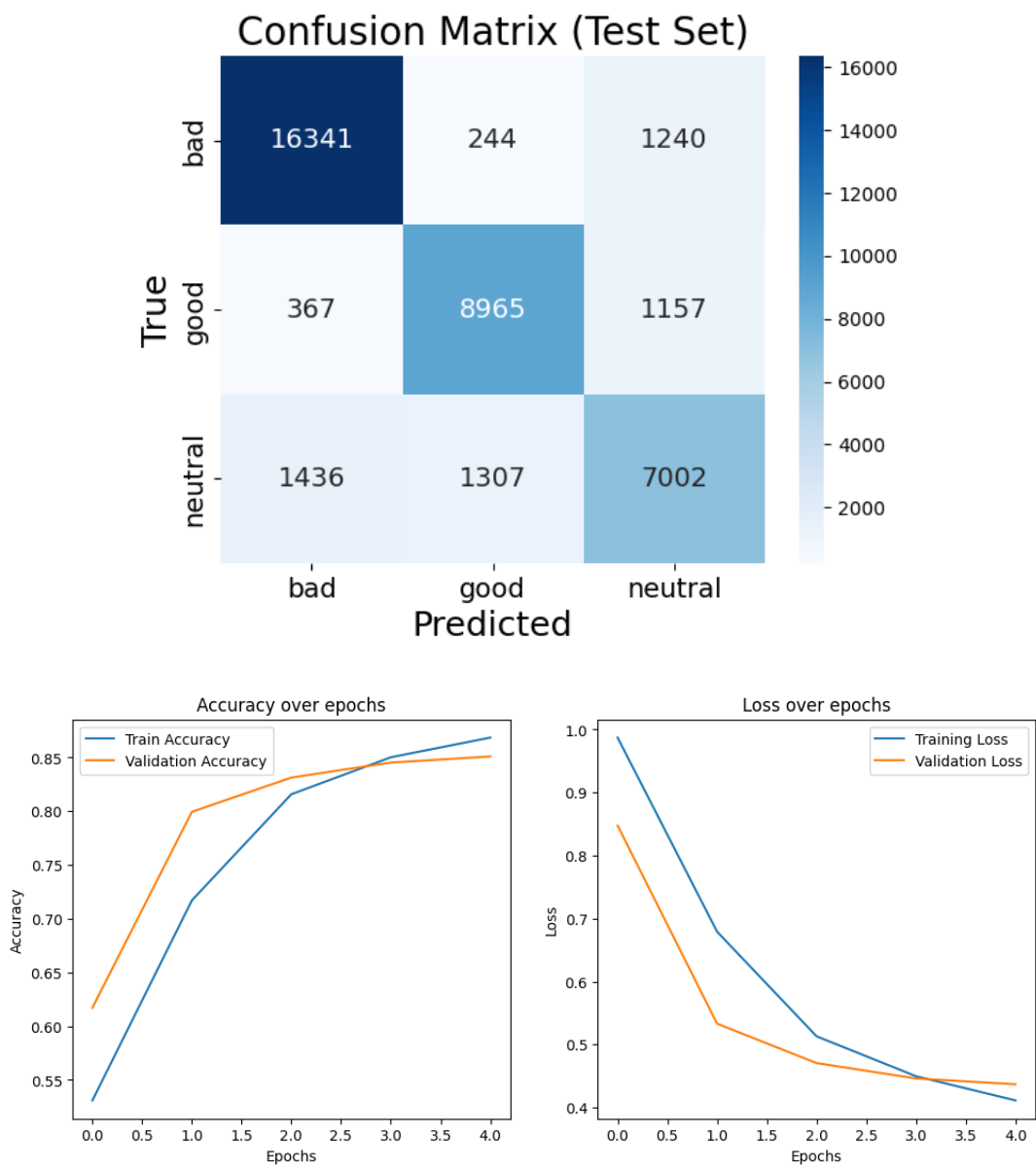
   accuracy          0.89      152234
  macro avg         0.88      0.88      0.88      152234
 weighted avg         0.89      0.89      0.89      152234

1190/1190 [=====] - 4s 3ms/step - loss: 0.4395 - accuracy: 0.8502
Test Accuracy: 85.02%
1190/1190 [=====] - 3s 3ms/step
Test F1 Score: 84.95
Test Precision: 84.90
Test Recall: 85.02
Classification Report for Test Data:
      precision    recall  f1-score   support

    bad         0.90      0.92      0.91      17825
    good         0.86      0.85      0.85      10489
   neutral         0.74      0.72      0.73       9745

   accuracy          0.85      38059
  macro avg         0.83      0.83      0.83      38059
 weighted avg         0.85      0.85      0.85      38059

```



2. *learning_rate=0.00005, epochs=5, batch_size=64:*

```

Train Accuracy: 87.81%
4758/4758 [=====] - 15s 3ms/step
Train F1 Score: 87.73
Train Precision: 87.69
Train Recall: 87.81
Classification Report for Train Data:
      precision    recall  f1-score   support

    bad         0.92      0.94      0.93      71657
    good         0.88      0.88      0.88      41695
   neutral         0.80      0.77      0.78      38882

   accuracy                0.88      152234
  macro avg         0.86      0.86      0.86      152234
 weighted avg         0.88      0.88      0.88      152234

1190/1190 [=====] - 4s 3ms/step - loss: 0.4514 - accuracy: 0.8438
Test Accuracy: 84.38%
1190/1190 [=====] - 4s 3ms/step
Test F1 Score: 84.26
Test Precision: 84.18
Test Recall: 84.38
Classification Report for Test Data:
      precision    recall  f1-score   support

    bad         0.89      0.92      0.91      17825
    good         0.85      0.85      0.85      10489
   neutral         0.74      0.70      0.72      9745

   accuracy                0.84      38059
  macro avg         0.83      0.82      0.82      38059
 weighted avg         0.84      0.84      0.84      38059

```

3. *learning_rate=0.00005, epochs=10, batch_size=64:*

```

Train Accuracy: 89.51%
4758/4758 [=====] - 16s 3ms/step
Train F1 Score: 89.50
Train Precision: 89.49
Train Recall: 89.51
Classification Report for Train Data:
      precision    recall  f1-score   support

    bad         0.94      0.94      0.94      71657
    good         0.90      0.89      0.89      41695
   neutral         0.82      0.81      0.81      38882

   accuracy                0.90      152234
  macro avg         0.88      0.88      0.88      152234
 weighted avg         0.89      0.90      0.90      152234

1190/1190 [=====] - 5s 4ms/step - loss: 0.4416 - accuracy: 0.8508
Test Accuracy: 85.08%
1190/1190 [=====] - 5s 4ms/step
Test F1 Score: 85.05
Test Precision: 85.03
Test Recall: 85.08
Classification Report for Test Data:
      precision    recall  f1-score   support

    bad         0.90      0.92      0.91      17825
    good         0.86      0.84      0.85      10489
   neutral         0.74      0.74      0.74      9745

   accuracy                0.85      38059
  macro avg         0.84      0.83      0.83      38059
 weighted avg         0.85      0.85      0.85      38059

```

4. *learning_rate=0.0001, epochs=5, batch_size=64:*

```

Train Accuracy: 90.43%
4758/4758 [=====] - 16s 3ms/step
Train F1 Score: 90.42
Train Precision: 90.42
Train Recall: 90.43
Classification Report for Train Data:
      precision    recall  f1-score   support

    bad         0.94      0.95      0.94       71657
    good         0.92      0.89      0.90       41695
   neutral         0.83      0.83      0.83       38882

 accuracy         0.90         0.90         0.90       152234
  macro avg         0.90      0.89      0.89       152234
 weighted avg         0.90      0.90      0.90       152234

1190/1190 [=====] - 5s 4ms/step - loss: 0.4427 - accuracy: 0.8460
Test Accuracy: 84.60%
1190/1190 [=====] - 4s 3ms/step
Test F1 Score: 84.55
Test Precision: 84.53
Test Recall: 84.60
Classification Report for Test Data:
      precision    recall  f1-score   support

    bad         0.89      0.92      0.91       17825
    good         0.87      0.83      0.85       10489
   neutral         0.73      0.72      0.73        9745

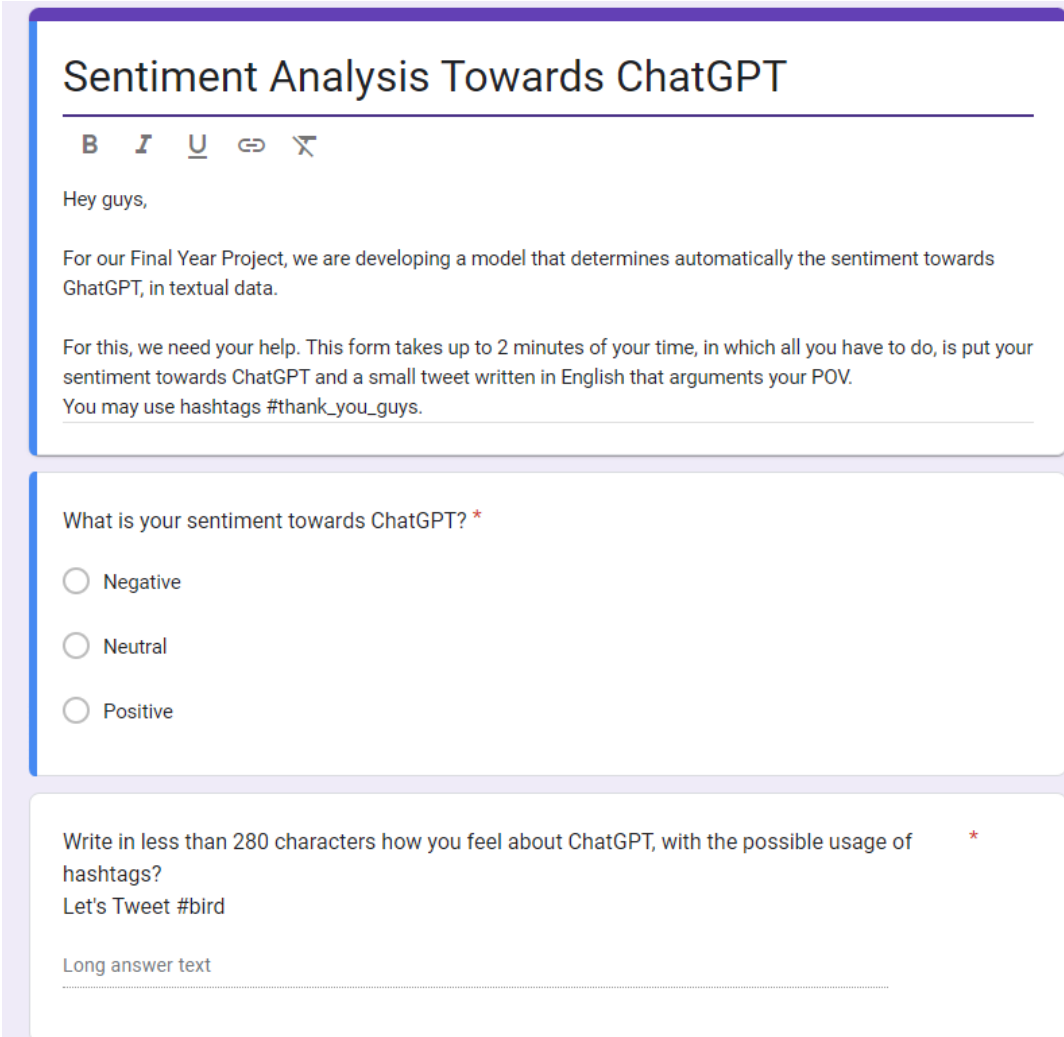
 accuracy         0.85         0.85         0.85       38059
  macro avg         0.83      0.83      0.83       38059
 weighted avg         0.85      0.85      0.85       38059

```

Appendix C

Google Form and Results

Form:



Sentiment Analysis Towards ChatGPT

Hey guys,

For our Final Year Project, we are developing a model that determines automatically the sentiment towards ChatGPT, in textual data.

For this, we need your help. This form takes up to 2 minutes of your time, in which all you have to do, is put your sentiment towards ChatGPT and a small tweet written in English that arguments your POV. You may use hashtags #thank_you_guys.

What is your sentiment towards ChatGPT? *

☐ Negative

☐ Neutral

☐ Positive

Write in less than 280 characters how you feel about ChatGPT, with the possible usage of hashtags? *

Let's Tweet #bird

Long answer text

Responses (26):

- I m in love with chat gpt #love - [Neutral]
- I believe that chatgpt is diminishing the capability of a person to use his brain. Instead of thinking how to write a mail we ask chatgpt, and how to write a code we ask chatgpt. It's the easy way and it's not giving us the ability to think for ourselves and develop our brain. - [Negative]

- #chatgpt is a helpful tool but still won't #help in some cases. Anyways it is so #important in our lives - [Positive]
- I utilized ChatGPT to craft this concise message. It's like having a knowledgeable bird whispering insights into your ear. #ChatGPT #bird #thank_you_guys - [Positive]
- Personally i really like it, it's impressive how it can process and generate responses this fast on such a wide range of topics #easierlife - [Positive]
- #friendly#useful #lifesaver #smart #creative #helpful #flow - [Positive]
- It's a tool I use to help me organize my thoughts or understand a certain task to do - [Positive]
- Good in helping us but it is not really reliable cz it can make mistakes - [Neutral]
- I don't hate it because sometimes it is actually useful ,but I don't love it, and I don't use it for anything other than summarizing or getting ideas because I don't trust its resources . It frequently gives wrong information and its large usage by people promotes the misreading - [Neutral]
- I don't really use chatgpt because I don't trust its answers - [Neutral]
- It's a perfect source of information when google doesn't understand your 2 sentence long inquiry, a debate partner that feeds into your ego #""That'sACleverArgument!"" , and a therapist that will tell you you're in a toxic relationship #DeluluIsNot-TheSolulu #FavAI #cutiepatootie" - [Positive]
- i love it #chatgpt #bird - [Positive]
- Chatgpt is essential to our needs in this generation no doubts but unfortunately it does have some drawbacks on jobs as a nunmer 1 problem - [Neutral]
- Very #helpful and gives #accurate #solutions and #answers! However the information is a bit #outdated since it's from 2021! - [Positive]
- I feel chatgpt very useful for general q/a concerning topics of some particular interests, given that one has to search for answers on gooogle, the tool gives direct answers, which saves time. However, it sometimes may misunderstand the input query and give undesired outputs. - [Neutral]
- I'll begin with one word: #helpful. Then, #impressive because this is one of the most powerful engine of its kind. But there is also a little bit of fear as it continue to developp other features. The creators themselves can't predict how far it could go. #SafetyPlizz - [Positive]
- c'est le progrès, le futur, une nouvelle manière de travailler et de voir le monde - [Positive]
- "i really find a good use in chat gpt, when I dont understand something in class, I usually type the questions that I needed an answer for, and the AI usually does the job. Also, when I struggle in maths, they explain all the exercice. Nevertheless, the tool shows some flaws" - [Positive]

- ChatGPT assists me in completing my coding projects, homework assignments, and exam preparation - [Positive]
- ChatGPT is a digital marvel, a beacon of knowledge and creativity. It's like having a wise companion always ready to lend a helping hand or spark an enlightening conversation. #ChatGPT #AI #Innovation - [Positive]
- ChatGPT is good if used as an assistant. People should not let it do the work they're supposed to do. Instead, they should use it like Google to get information and then combine the output with their knowledge. Overly depending on ChatGPT will make us less depending on ourselves. - [Neutral]
- I find ChatGPT helpful and reliable, like a trusted friend always ready to lend a hand. It's amazing how it understands and responds to my queries - [Positive]
- ChatGPT is a program made to answer people's questions and help them with their tasks it's a way so have a summarise of what they need in a very rapid way it lead to students specifically to use it during their projects but it led to students becoming lazy in their searches - [Neutral]
- Chatgpt is ruining our lives. It will leave soon enough leave us with no jobs #AITakeover #Chatgpt #annoying - [Negative]
- I think it helps a lot of people with writing difficulties, and in some ways it will give people some inspiration in work or study - [Positive]
- ChatGpt is not only a bad technology, it is very mind numbing... everyone is using it as an easy way out which is very annoying - [Negative]

Graphs:

What is your sentiment towards ChatGPT?

26 responses

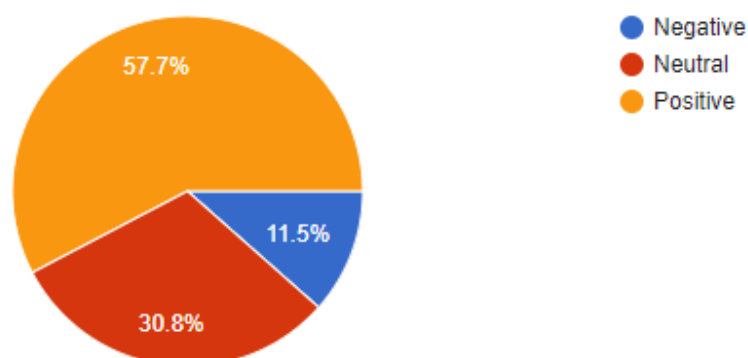


FIGURE C.1: Sentiment Distribution of Responses

Bibliography

- Ahmad, Munir et al. (Apr. 2017). "Machine Learning Techniques for Sentiment Analysis: A Review". In: *INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING* 8, pp. 2045–7057.
- Aqlan, Ameen, Dr. Manjula Bairam, and R Lakshman Naik (Jan. 2019). "A Study of Sentiment Analysis: Concepts, Techniques, and Challenges". In: pp. 147–162. ISBN: 978-1-4939-9044-3. DOI: [10.1007/978-981-13-6459-4_16](https://doi.org/10.1007/978-981-13-6459-4_16).
- Bonnet, Alexandre (2023). "Accuracy vs. Precision vs. Recall in Machine Learning: What is the Difference?" In: URL: <https://encord.com/blog/classification-metrics-accuracy-precision-recall/>.
- Bordoloi, Monali and Saroj Biswas (Mar. 2023). "Sentiment Analysis: A Survey on Design Framework, Applications and Future Scopes". In: *Artificial Intelligence Review* 56. DOI: [10.1007/s10462-023-10442-2](https://doi.org/10.1007/s10462-023-10442-2).
- Dang, Nhan Cach, María N. Moreno García, and Fernando de la Prieta (2020). "Sentiment Analysis Based on Deep Learning: A Comparative Study". In: *CoRR* abs/2006.03541. arXiv: [2006.03541](https://arxiv.org/abs/2006.03541). URL: <https://arxiv.org/abs/2006.03541>.
- Devika, M.D., C. Sunitha, and Amal Ganesh (2016). "Sentiment Analysis: A Comparative Study on Different Approaches". In: *Procedia Computer Science* 87. Fourth International Conference on Recent Trends in Computer Science Engineering (ICRTCSE 2016), pp. 44–49. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2016.05.124>. URL: <https://www.sciencedirect.com/science/article/pii/S187705091630463X>.
- Diego Lopez, Yse (2021). "NLP — Text Vectorization". In: URL: <https://lopezyse.medium.com/nlp-text-vectorization-e472a3a9983a>.
- Drus, Zulfadzli and Haliyana Khalid (2019). "Sentiment Analysis in Social Media and Its Application: Systematic Literature Review". In: *Procedia Computer Science* 161. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia, pp. 707–714. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2019.11.174>. URL: <https://www.sciencedirect.com/science/article/pii/S187705091931885X>.
- Gautam, Hariom (2020). "Word Embedding: Basics". In: URL: <https://medium.com/@hari4om/word-embedding-d816f643140>.
- K, Bharath (2023). "Introduction to Deep Neural Networks". In: URL: <https://www.datacamp.com/tutorial/introduction-to-deep-neural-networks>.
- Kerner, Sean Michael (2024). "What are large language models (LLMs)?" In: URL: <https://www.techtarget.com/whatis/definition/large-language-model-LLM>.
- Kok, Mihai Rotaru Kasper (n.d.). "SEVEN LIMITATIONS OF LARGE LANGUAGE MODELS (LLMS) IN RECRUITMENT TECHNOLOGY". In: (). URL: <https://www.textkernel.com/learn-support/blog/seven-limitations-of-llms-in-hr-tech/>.
- Mandelbaum, Amit and Adi Shalev (2016). "Word Embeddings and Their Use In Sentence Classification Tasks". In: URL: <https://arxiv.org/pdf/1610.08229>.

- Nguyen, Eric (2014). "Text Mining and Network Analysis of Digital Libraries in R". In: URL: <https://www.sciencedirect.com/topics/computer-science/inverse-document-frequency>.
- Paul Penfield, Jr (2003). "Principle of Maximum Entropy". In: URL: <https://mtlsites.mit.edu/Courses/6.050/2003/notes/chapter10.pdf>.
- Rambocas, Meena (Apr. 2013). "Marketing research: The role of sentiment analysis". In: *FEP WORKING PAPER SERIES*.
- Raphael Couronné, Philipp Probst Anne-Laure Boulesteix (2018). "Random forest versus logistic regression: a large-scale benchmark experiment". In: URL: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2264-5>.
- Robinson, Derek W. (2008). "Entropy and Uncertainty". In: URL: <https://www.mdpi.com/1099-4300/10/4/493>.
- Salinca, Andreea (2015). "Business Reviews Classification Using Sentiment Analysis". In: *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 247–250. DOI: [10.1109/SYNASC.2015.46](https://doi.org/10.1109/SYNASC.2015.46).