People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 Mai 1945 - Guelma
Faculty of Mathematics, Computer Science and Material Science
Computer Science Department

Master's thesis

Branch: Computer Science

Option: ICST

Topic:

Comparison of NLP Techniques for sentiment analysis on social data (application case: war in GAZA)

Presented By

ALLAL Younes

In Front of The Jury

Dr. BOUGHAREB Djalila	President
Pr. KOUAHLA Mohamed Nadjib	Supervisor
Dr. BENAMIRA Adel	Reviewer

June 2024

ABSTRACT

The field of sentiment analysis (SA) has experienced a significant resurgence with the advancement of artificial intelligence (AI) techniques in recent years. Its use has been widely associated with the analysis of public opinions. Given the global attention on the war in Gaza, this study aims to apply and compare various natural language processing (NLP) techniques for sentiment analysis on public comments from social media related to this conflict. Different classification approaches were employed, including traditional machine learning, deep learning, and transfer learning. The results indicated that the majority of comments expressed negative sentiments towards the war. Notably, the DistilBERT classifier achieved the highest classification accuracy at 89%, slightly outperforming the LSTM model, which achieved an accuracy of 88%. The findings of this study will serve to inform and stimulate future research in this evolving field.

Keywords: Gaza, Sentiment Analysis (SA), Social Media, Public Opinion, Text Classification, Machine Learning (ML), Deep Learning (DL), Transfer Learning (TL).

ملخص

لقد شهد مجال تحليل المشاعر (SA) انتعاشًا كبيرًا مع تطور تقنيات الذكاء الاصطناعي (AI) في السنوات الأخيرة. وقد ارتبط استخدامه بشكل واسع بتحليل الآراء العامة. نظرًا للاهتمام العالمي بالحرب في غزة، تهدف هذه الدراسة إلى تطبيق ومقارنة تقنيات معالجة اللغة الطبيعية (NLP) المختلفة لتحليل المشاعر من التعليقات العامة المتواجدة على وسائل التواصل الاجتماعي المتعلقة بهذا الصراع. تم استخدام نهج تصنيف مختلفة، بما في ذلك التعلم الآلي التقليدي، والتعلم العميق، والتعلم التحويلي. أشارت النتائج إلى أن غالبية التعليقات أعربت عن مشاعر سلبية تجاه الحرب. ومن الجدير بالذكر أن مصنف DistilBERT حقق أعلى دقة في التصنيف بنسبة 89%، متفوقًا بشكل طفيف على نموذج LSTM الذي حقق دقة بنسبة 88%. ستساعد نتائج هذه الدراسة في إرشاد وتحفيز البحوث المستقبلية في هذا المجال المتطور.

الكلمات المفتاحية: غزة، تحليل المشاعر (SA)، وسائل التواصل الاجتماعي، آراء الجمهور، تصنيف النصوص، التعلم الآلي (ML)، التعلم العميق (DL)، التعلم التحويلي (TL).

RÉSUMÉ

Le domaine de l'analyse des sentiments a connu un regain significatif avec l'avancement des techniques d'intelligence artificielle ces dernières années. Son utilisation a été largement associée à l'analyse des opinions publiques. Compte tenu de l'attention mondiale portée à la guerre à Gaza, cette étude vise à appliquer et comparer diverses techniques de traitement de langage naturel pour l'analyse des sentiments sur les commentaires publics des réseaux sociaux liés à ce conflit. Différentes approches de classification ont été employées, y compris l'apprentissage automatique traditionnel, l'apprentissage profond et l'apprentissage par transfert. Les résultats ont indiqué que la majorité des commentaires exprimaient des sentiments négatifs envers la guerre. Notamment, le classificateur DistilBERT a atteint la précision de classification la plus élevée avec 89%, surpassant légèrement le modèle LSTM, qui a atteint une précision de 88%. Les conclusions de cette étude serviront à informer et stimuler les recherches futures dans ce domaine en évolution.

Mots-clés : Gaza, Analyse des Sentiments, Réseaux Sociaux, Opinion Publique, Classification de Texte, Apprentissage Automatique, Apprentissage Profond, Apprentissage par Transfert.

إهداء

قال تعالى:

ذَٰلِكَ بِأَنَّ اللَّهَ مَوْلَى الَّذِينَ آمَنُوا وَأَنَّ الْكَافِرِينَ لَا مَوْلَىٰ لَهُمْ ﴿١١﴾

أهدي هذا العمل إلى إخواننا المسلمين الصابرين في غزة،
سائلا المولى عزّ وجل لهم النّصر والفرج القريب.

شكر و عرفان

أشكر الله رب العالمين الذي خلق و هدى و سدّد الخطى فخرج هذا العمل بعونه و توفيقه نحمده حمدًا كثيرا. والصلاة والسلام على سيدنا و حبيبنا و قائدنا و قدوتنا محمد عليه أفضل الصّلاة وأزكى التّسليم.

أمّا بعد..

أودّ أن أعبر عن امتناني العميق لمشرفي: السيد كواحلة محمد نجيب على توجيهاته القيمة ودعمه المستمر و تشجيعه طوال فترة بحثي. لقد كان لخبراته ورؤاه الدور الفعال في إنجاز هذا العمل.

كما أُعرب عن امتناني لأعضاء لجنة أطروحتي: السيدة بوغارب جليلة والسيد بن عميرة عادل، على تخصيصهم الوقت الكافي لتقييم عملي، سيكون لنقدكم البنّاء ونصائحكم الدّور الفعّال في تطوير عملي بإذن الله تعالى.

كما أدين بالشكر الجزيل و الامتنان الكبير لعائلتي: لوالديَّ اللّذين صبرا عليّ في أضعف مراحلي، لأجدادي من أحسنوا تربيتي، لخالي و خالتي من كانا سندًا لي طوال فترة حياتي، لإخوتي من كانوا سببا في سعادتي، لبنتَي خالتي اللّتين رسمتا البسمة على مُحيّاي، لعمّي من أخذ بيدي و عمّاتي و جميع أفراد عائلتي صغيرهم و كبيرهم، للّذينَ كان صبرهم ودعمهم الثابت أساسًا لي. لقد كان إيمانكم بي مصدر تحفيز دائم لي.

شكرٌ خاص لأصدقائي، لزملاء مدرستي وثانويتي و من أمضيت معهم أفضل فترات حياتي، لزملائي في قسم الإعلام الآلي التي جعلت صداقتهم الطيبة هذه الرحلة لا تنسى.

وأخيرًا، أنا ممتن لكل من ساهم من قريب أو من بعيد في إنجاح هذا العمل..

ممتن لدعائكم، لمساعدتكم و تشجيعكم.

بارك الله فيكم و جزاكم الله كل خير.

CONTENTS

Abstract **I**

Dedication **IV**

Acknowledgments **V**

List of Figures **IX**

List of Tables **X**

List of Acronyms **XI**

List of Code Snippets **XIII**

General Introduction **1**

I Theoretical Concepts **4**

1 Introduction 4

2 Sentiment Analysis 4

2.1 Definition 4

2.2 Sentiment Analysis and Emotion Recognition 5

2.3 Data Types 5

2.4 Applications 6

3 Key Stages in SA Process 7

3.1 Text Representation 7

3.2 Sentiment Classification 9

3.3 Performance Evaluation 13

4 Related Works 14

5 Conclusion 18

II Conception **19**

1 Introduction 19
2 General Architecture 19
3 Methodology 20
3.1 Data Collection 20
3.2 Data Annotation (VADER) 21
3.3 Data Preprocessing 22
3.4 Feature Extraction 23
3.5 Classification 24
3.6 Evaluation 25
3.7 Deployment 25
4 Conclusion 26

III Implementation and Results **27**

1 Introduction 27
2 Work Environment 28
2.1 Google Colab 28
2.2 Kaggle 28
3 Programming Language and Libraries 28
3.1 Python 28
3.2 Libraries 28
4 Implementation 29
4.1 Data Collection 29
4.2 Data Annotation 30
4.3 Data Preprocessing 31
4.4 Data Set Splitting 33
4.5 Feature Extraction 33
4.6 Classification 35
4.7 Evaluation 37
4.8 Deployment 37
5 Obtained Results 38
5.1 Collected Data 38
5.2 Labeled Data 39
5.3 Data After Preprocessing 41
5.4 Training Set and Testing Set 42
5.5 Classification 43
5.6 Deployment 44
6 Additional Experiments 45

6.1 Annotation: AFINN 46
6.2 Preprocessing: Removing Stop Words 47
6.3 Classification: Hyperparameter Tuning for ML 49
7 Conclusion 50

General Conclusion **51**

Bibliography **53**

Webography **58**

A Dataset **59**

B Hyperparameter Tuning **60**

LIST OF FIGURES

I.1 Data types 5
I.2 Different Approaches of Sentiment Classification 9
I.3 Supervised Learning vs Semi-Supervised Learning vs Unsupervised Learning [41] 11
I.4 The DistilBERT model architecture and components [2] 13

II.1 General Architecture 20
II.2 Data Collection 21
II.3 Data Preprocessing 23
II.4 Architecture of LSTM Model 24
II.5 Architecture of CNN Model 25

III.1 Number of Comments in Each Subreddit 39
III.2 Wordcloud of The Most Frequent Words 39
III.3 Number of Comments by Sentiment 40
III.4 Web Interface 45
III.5 Web Interface (Testing with DistilBERT) 45
III.6 VADER vs AFINN 46

A.1 Dataset (https://github.com/unus-all/sentiment-analysis) 59

B.1 Results of GridSearchCV 60

LIST OF TABLES

I.1 Bag of Words Example 8
I.2 Some Advantages and Disadvantages of Different Approaches 13
I.3 Taxonomy 18

III.1 Sample of The Labeled Comments 41
III.2 Before and After Preprocessing 42
III.3 Data Distribution Between Training and Testing Sets 43
III.4 ML-Based Models 43
III.5 Optimal Configuration 44
III.6 DL-Based Models 44
III.7 Fine-tuned DistilBERT 44
III.8 ML-Based Models (AFINN) 47
III.9 DL-Based Models (AFINN) 47
III.10 Fine-tuned DistilBERT (AFINN) 47
III.11 DL-Based Models (After Removing Stop Words) 48
III.12 Fine-tuned DistilBERT (After Removing Stop Words) 48
III.13 Best Hyperparameters 50
III.14 ML-Based Models (After Hyperparameter Tuning) 50

ACRONYMS

ADA Adaptive Boosting.

BN Bayes Network.

BoW Bag of Words.

CNN Convolutional Neural Networks.

DL Deep Learning.

DT Decision Tree.

ETC Extra Trees Classifier.

GNB Gaussian Naive Bayes.

KNN K-Nearest Neighbour.

LinearSVC Linear Support Vector Classification.

LR Logistic Regression.

LSTM Long Short-Term Memory.

ML Machine Learning.

MNB Multinomial Naive Bayes.

NB Naive Bayes.

NLP Natural Language Processing.

NLTK Natural Language Toolkit.

NN Neural Networks.

PRAW Python Reddit API Wrapper.

RF Random Forest.

RRL Ripper Rule Learning.

SA Sentiment Analysis.

SGD Stochastic Gradient Descent.

SVM Support Vector Machine.

TF-IDF Term Frequency – Inverse Document Frequency.

TL Transfer Learning.

VADER Valence Aware Dictionary for sEntiment Reasoning.

XGB eXtreme Gradient Boosting.

LIST OF CODE SNIPPETS

III.1 Data Collection (PRAW) . . . 30
III.2 Data Annotation (VADER) . . . 31
III.3 Data Preprocessing . . . 32
III.4 Padding . . . 32
III.5 Data Splitting . . . 33
III.6 TF-IDF and BoW . . . 33
III.7 GloVe . . . 34
III.8 Word2Vec . . . 34
III.9 Embedding Matrix . . . 34
III.10 Training a ML-Based Classifier (e.g, MNB) . . . 35
III.11 Initializing the LSTM Model . . . 36
III.12 Initializing the CNN Model . . . 36
III.13 Training a DL-Based Classifier . . . 37
III.14 Performance Evaluation . . . 37
III.15 Flask - General Implementation . . . 38
III.16 Text Annotation (AFINN) . . . 46
III.17 Removing Stop Words . . . 48
III.18 Hyperparameter Tuning (e.g, MNB) . . . 49

GENERAL INTRODUCTION

Sentiment analysis (SA), also referred to as opinion mining, is the field of study focused on examining and interpreting people's opinions, sentiments, attitudes, and emotions toward various entities such as products, services, events, and individuals [23]. While this field has a substantial history, it currently stands out as one of the fastest-growing research domains within computer science, propelled by significant advancements in artificial intelligence.
The relevance of sentiment analysis extends across various domains, including medicine for analyzing patient psychology and marketing for gauging consumer sentiment towards products and enhancing them. Particularly prominent is its application in politics, where sentiment analysis is frequently utilized to gauge public opinion towards political figures and societal issues, thereby influencing policy decisions and public perceptions.

Historically, political opinion was predominantly shaped by the content reported in newspapers and broadcast on television and radio. This paradigm provided a significant advantage to politicians and those in power, allowing them to disseminate their ideologies to the public unchallenged. However, this monopolization of information dissemination precluded widespread public debate and criticism of governmental policies. This dynamic persisted for an extended period until the advent of social media in the past two decades. The emergence of social media platforms revolutionized the landscape of political discourse by empowering the public to actively engage in debates, including those of a political nature. Criticisms of specific policies, once confined to private conversations, now had the potential to be aired on a global scale. The widespread sharing of dissenting opinions not only facilitated robust public debate but also posed significant threats to the interests of entire companies and organizations, thereby altering the balance of power in public discourse.
One of the major political situations that has dominated the global stage in recent months is the ongoing situation in Gaza and the brutal war waged by Israeli occupation forces, which has continued for almost nine months. Since the first Nakba in 1948, the Palestinian cause has never garnered as much attention, almost disappearing under the guise of normalization with

the occupation.
For many years, the Zionist narrative has dominated Western media by adopting a victim mentality, portraying Zionists as innocent people needing to defend themselves against what they call savage terrorists. This need for such narratives increased, especially after the Palestinian resistance's attack on the occupied Palestinian territories on October 7th, using it as an excuse for killing and destruction. This narrative's control extended beyond the general public, as even at the war's onset, the American administration blindly relied on the statements of Zionist leaders, presenting them to the American people as absolute truth. However, with each passing day of the war and aggression on the Gaza Strip, this situation has changed. According to an article from Al-Jazeera [21], the Israeli entity has already lost the war of public opinion, which it had previously dominated, especially after killing, wounding, and starving tens of thousands of people, most of whom were women and children. This created a gap between reality and the portrayal of its army as the most moral in the world, disregarding the decisions of the International Criminal Court and the UN Security Council, and ignoring the calls of international human rights, health, and humanitarian organizations.
The global public's participation in sharing what is happening inside Gaza on social media has contributed to a global uprising once the true child-killer was revealed. Western people's trust in their governments has become nearly nonexistent, reflected in hundreds of thousands of people participating in protests over the past months against their countries' handling of the war. This led to universities and even countries severing their relationships with and support for the Israeli entity. This impact extended beyond educational and administrative institutions to commercial companies. Due to campaigns to boycott products supporting occupation forces, which began and spread through social media, major global companies like Starbucks and McDonald's suffered losses amounting to billions of dollars, leading them to close some branches and lay off workers [27, 48].

In light of these circumstances and the significant influence of public opinion on social media platforms on the course of this conflict, our study aims to compare various NLP techniques for sentiment analysis and determine the polarity of people's opinions on the war in Gaza. People's sentiments and their polarity are automatically classified using different but somewhat related methods. One of the simplest and earliest methods relies on lexicons, where each word has an emotional value. Despite the ease of applying this technique, its accuracy is limited due to its inability to understand context and comprehend sarcastic sentences. Another method is using traditional machine learning techniques, which primarily rely on studying features extracted from texts using specific techniques. A more advanced method is training deep learning classifiers to recognize patterns in sentiment identification from texts. The most popular method today is transfer learning; the ability to fine-tune pre-trained models on billions of data points and adapt them to context based on their acquired expertise is a labor, cost, and time-saving approach, often more accurate than its predecessors.

Through this study, we seek to find answers to the following research questions:

- **RQ1:** How does the accuracy of labeling collected data affect the efficiency of the previously mentioned classification techniques?
- **RQ2:** Are there any difficulties in processing the colloquial language used on social media?
- **RQ3:** Among the aforementioned classification techniques, which one will be the most efficient?

This thesis is organized into three main chapters. In the first chapter, we discuss the theoretical concepts surrounding sentiment analysis, including definitions, classification methods, and related research. In the second chapter, we present our research methodology and workflow, supported by illustrative diagrams. The third chapter provides a detailed explanation of the practical aspect of our methodology, mentioning the tools and platforms used, along with code snippets, and concluding with the obtained results and additional experiments conducted to gain a broader understanding of the problem.
Finally, the three chapters are concluded with a general summary that recaps the key points of this thesis, presenting some ideas for future work.

CHAPTER I

THEORETICAL CONCEPTS

Contents

1	**Introduction**	4
2	**Sentiment Analysis**	4
3	**Key Stages in SA Process**	7
4	**Related Works**	14
5	**Conclusion**	18

1 Introduction

Sentiments constitute a significant portion of an individual's identity and can play a pivotal role in decision-making processes. Therefore, the ability to identify and analyze them is a crucial skill in many contexts.

This chapter will examine the field of sentiment analysis, including the representation of sentiments, the applied fields, and the automatic methods of this analysis. It will also cite previous research in this area.

2 Sentiment Analysis

2.1 Definition

Sentiment analysis (SA), also referred to as opinion mining, is the field of study focused on examining and interpreting people's opinions, sentiments, attitudes, and emotions toward various entities such as products, services, events, and individuals [23]. It provides a valuable

opportunity to explore the mindsets of individuals towards a particular situation and study them from different perspectives.

2.2 Sentiment Analysis and Emotion Recognition

While sentiments and emotions may often be seen as synonyms, their applications in the field of natural language processing (NLP) may differ slightly. Sentiment analysis (SA) primarily focuses on the classification of the overall sentiment, which is typically categorized into polarity classes (positive, negative, or neutral). In contrast, emotion detection is a process that aims to identify and categorize data into specific emotional states, such as happiness, sadness, anger, and fear, providing a more nuanced understanding of the emotional content.
However, emotion recognition can often be considered a subtask of polarity detection [9]. Moreover, various applications are more appropriately suited to emotion recognition than to polarity detection, and vice versa.
To mitigate any conceptual conflicts, we will employ both terms in the subsequent sections, choosing the one that best fits each specific application.

2.3 Data Types

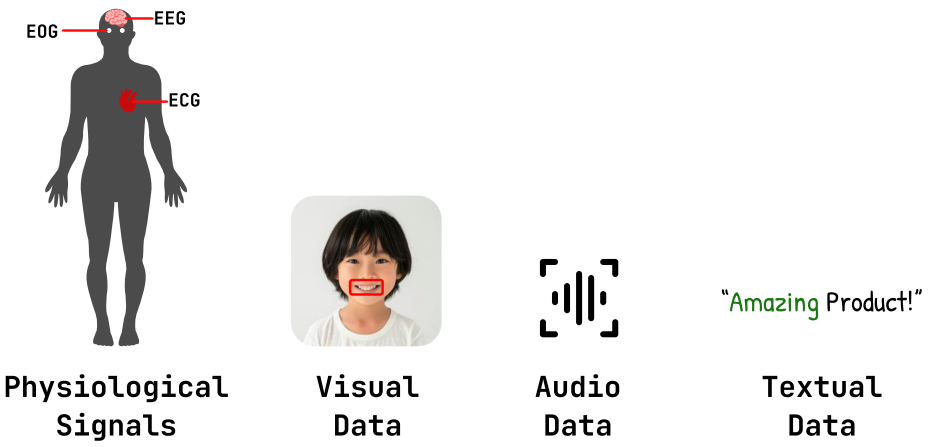

Figure I.1: Data types

2.3.1 Textual Data

Sentiment analysis is typically associated with textual data, and it remains the most prevalent and well-researched area. The process of textual sentiment analysis involves the transformation of written language into a format that can be analyzed to determine the sentiment expressed within the text. This method has been employed extensively across a range of platforms, especially in social media (e.g, [52]).

2.3.2 Visual Data

Emotion recognition from images frequently rely on the detection and recognition of faces. Recent advancements in AI have enabled more accurate identification of facial expressions, which can be analyzed to infer emotions [10, 19]. The process often necessitates the use of complex algorithms, which are capable of detecting subtle alterations in facial features and thereby enabling the classification of emotional states on an individual basis.

2.3.3 Audio Data

Recognizing emotions from audio typically involves the assessment of the tone of voice by examining pitch, volume, and speech rate [22]. Additionally, audio can be converted into text, followed by the application of textual sentiment analysis methods.

2.3.4 Physiological Signals

A multitude of physiological signals, derived from sensors, may be utilized for the recognition of emotional states, including:

- **Electroencephalography (EEG):** Measures electrical activity in the brain, which can vary with different emotional states [44, 17].
- **Electrocardiography (ECG):** Monitors heart rate and rhythm, which are influenced by emotional reactions [42].
- **Electrooculography (EOG):** Tracks eye movements and pupil dilation can be used to infer emotional changes [47].

These signals change in accordance with emotional changes, without being under the conscious control of the human being. As a consequence, they represent an important type of data for many operations, including sentiment analysis.
For further reading, this paper [16] offers a comprehensive review of different sensors and their use in recognizing human emotions.

2.4 Applications

The popularity of sentiment analysis is widespread, and its applications go beyond the boundaries of one field:

Politics: The utilization of applications in this field is frequently employed by decision-makers. For instance, during election periods, the use of such applications can facilitate the preparation of election campaigns by enabling the study of individuals' political orientations and demands. Furthermore, during periods of conflict and other significant political events,

these applications can be employed to gain insight into the public's mentality and, in certain instances, to influence it, with the intention of instilling a specific ideology [11, 26].

Healthcare: As a recent example, we may cite the period of the COVID-19 crisis four years ago, during which the majority of people were confined to their homes in front of social media. This contributed to a great deal of research into people's emotions towards the disease or the vaccination [12, 52, 38]. In addition to textual data, visual data was also employed in research, such as how face masks (which were widely used during the pandemic) affected emotion recognition [18, 25]. Furthermore, data derived from sensors attached to the brain have also been subjected to extensive research, particularly in the context of studying the psychology of patients. [36]

Marketing and Service Optimization: SA can help sellers and business owners in particular, for example by studying people's opinions towards a particular product, how their target audience perceives it and what elements should be improved. From there, it is possible to derive their desires and find out what will make them happy, which will help in decision-making and improve reputation. Such applications can be found in hotels [30], restaurants [5], etc.

E-Learning: It has significantly benefited from sentiment analysis applications in studying the psychology of students and their response to distance education [54].

Finance and Stock Market: By assessing market sentiment through news articles, social media posts, and financial reports, SA can help predict stock price movements and market trends [46, 29].

Entertainment and Media: Understanding audience reactions to movies, TV shows, music, and other media content through social media and review analysis can help in content creation and marketing strategies [56].

3 Key Stages in SA Process

To build an effective automatic classifier for textual sentiment analysis, several key stages are essential: data representation, choosing the appropriate classification approach, and evaluating the model.

3.1 Text Representation

Before proceeding with sentiment analysis, it is imperative to apply a text representation technique. This step is essential because ML, DL, and TL, require numerical representations to process textual data effectively. Some of the most commonly used techniques include:

3.1.1 Bag of Words (BoW)

The Bag of Words (BoW) technique is a widely used method for representing text as numerical values, where each word is characterized by its frequency of occurrence within a document. This approach disregards the structure and order of words, which is why it is referred to as a "bag".

Example: Let's have these two sentences (documents):

- **D1:** free palestine free palestine
- **D2:** stop the genocide in gaza

The BoW representation is illustrated in the table I.1

Documents	free	palestine	stop	the	genocide	in	gaza
D1	2	2	0	0	0	0	0
D2	0	0	1	1	1	1	1

Table I.1: Bag of Words Example

3.1.2 Term Frequency – Inverse Document Frequency (TF-IDF)

TF-IDF is a Natural Language Processing (NLP) technique that quantifies the importance of a given word in a document and in a corpus of documents.
A word w_i is important to a document d_i if its appearance frequency in d_i is high, while its appearance frequency in other documents is low. This results in the word w_i being present in the specific document with greater frequency than in many other documents, making it significant for d_i.

- **Term Frequency (TF):** It is a statistical measure that quantifies the frequency of occurrence of a given term or word in a document.

$$TF(t,d) = \frac{\text{Number of occurences of term } t \text{ in a document } d}{\text{Total number of terms in the document } d}$$

- **Inverse Document Frequency (IDF):** It measures the importance of the term across a corpus.

$$IDF(t) = \log \frac{\text{Total number of documents in the corpus}}{\text{Number of documents with term } t \text{ in them}}$$

The TF-IDF is the product of these two terms:

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

3.1.3 Word Embeddings

Word embeddings are a way of representing words or sentences with vectors. They are distributional representations based on the distributional hypothesis (a set of statements attributed to different authors [1]), which says that words with similar meanings occur in similar contexts. Thus, words with similar meanings should have similar vector representations.
Two of the most popular distributed representation methods are:

- **Word2Vec [28]:** It takes as its input a large corpus of words and produces a vector space for each word. The two types of architecture that have are Continuous Bag of Words (CBOW) model and Skip-gram model.
- **GloVe [34]:** is another technique to generate word embeddings by constructing a large matrix of co-occurrence information and then counting each word and how often we see that word in a given context in a large corpus.

3.2 Sentiment Classification

There are numerous methods for identifying sentiments, particularly in the context of textual analysis, which is the focus of our research. The diagram in figure I.2 illustrates the various methods of sentiment analysis.

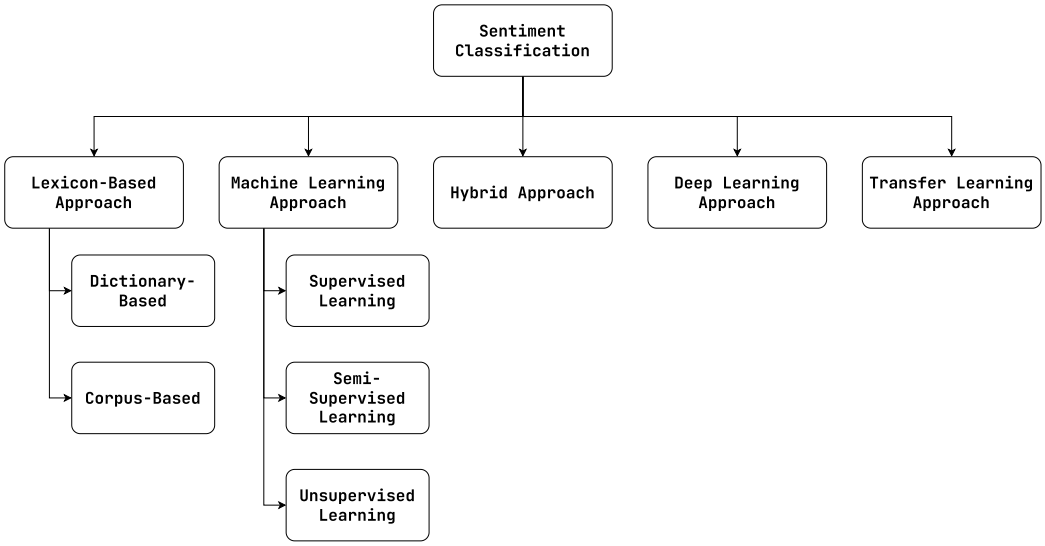

Figure I.2: Different Approaches of Sentiment Classification

3.2.1 Lexicon-based Approach

Dictionary-based Approach The most common and easiest to implement approach for sentiment analysis is to create a predefined list of words with assigned sentiment scores. For

[1] One of the most frequently cited versions appears to be that of: Rubenstein and Goodenough [39]

instance, positive words would receive a positive number, while negative words would receive a negative number.
To create such a list, a set of words from a specific domain is initially collected manually and given sentiment scores. This list is then expanded using online resources such as dictionaries or the popular WordNet network by adding synonyms and antonyms.
This list can then be employed to ascertain the sentiment polarity of a text by aggregating the sentiment scores of its constituent words.

Corpus-based Approach This approach, often referred to as corpus-based because of its reliance on large textual data, starts with a predefined set of sentiment terms and their orientations. It then analyzes syntactic and similarity patterns to identify sentiment tokens and their orientations within a large corpus [55].

3.2.2 Machine Learning Approach

In machine learning, the primary approaches are categorized into three major types based on the labeling status of the data.

Supervised Learning Supervised learning represents a dominant approach in machine learning (ML), particularly in the context of sentiment analysis. The process entails the training of algorithms (models) on labeled datasets. The models are able to identify patterns within the data, which enables them to accurately predict the sentiment of new, unseen data. Some popular algorithms are:

- **Multinomial Naive Bayes (MNB):** MNB is a very popular and efficient ML algorithm that is based on the popular Bayes' theorem. It uses probabilities to categorize text data. It's particularly useful for analyzing data where features are represented by word counts or how often events occur. This makes it a good choice for many tasks in NLP.
  While the multinomial distribution typically necessitates integer feature counts, in practice, fractional counts such as TF-IDF may also work.
- **Linear Support Vector Classification (LinearSVC):** a supervised machine learning algorithm similar to SVM, we can say that is a variant of the SVM which aims to find a hyperplane that separates classes with maximum margin. The main difference between them is the choice of the default loss function and penalties.
- **Logistic Regression (LR):** another supervised machine learning algorithm for classification tasks. Its objective is to determine the probability that an instance belongs to a specific class.

- **Decision Tree (DT):** A machine learning algorithm which used for both classification and regression tasks. It has a hierarchic, tree-like layout, consisting of a root, branches, internal nodes, and leaves.

Unsupervised Learning Unlike supervised learning stated previously, unsupervised learning does not require labeled data sets (it does not need human supervision). And in the context of sentiment analysis, this can include techniques such as clustering. Although this approach is not as widely utilized as supervised algorithms, several studies have been conducted employing this method [24].

Semi-Supervised Learning In ML, semi-supervised learning is an approach that utilizes both labeled and unlabeled data in the training phase. Most popular techniques in semi-supervised learning approach are self training, co-training and graph-based labeling.

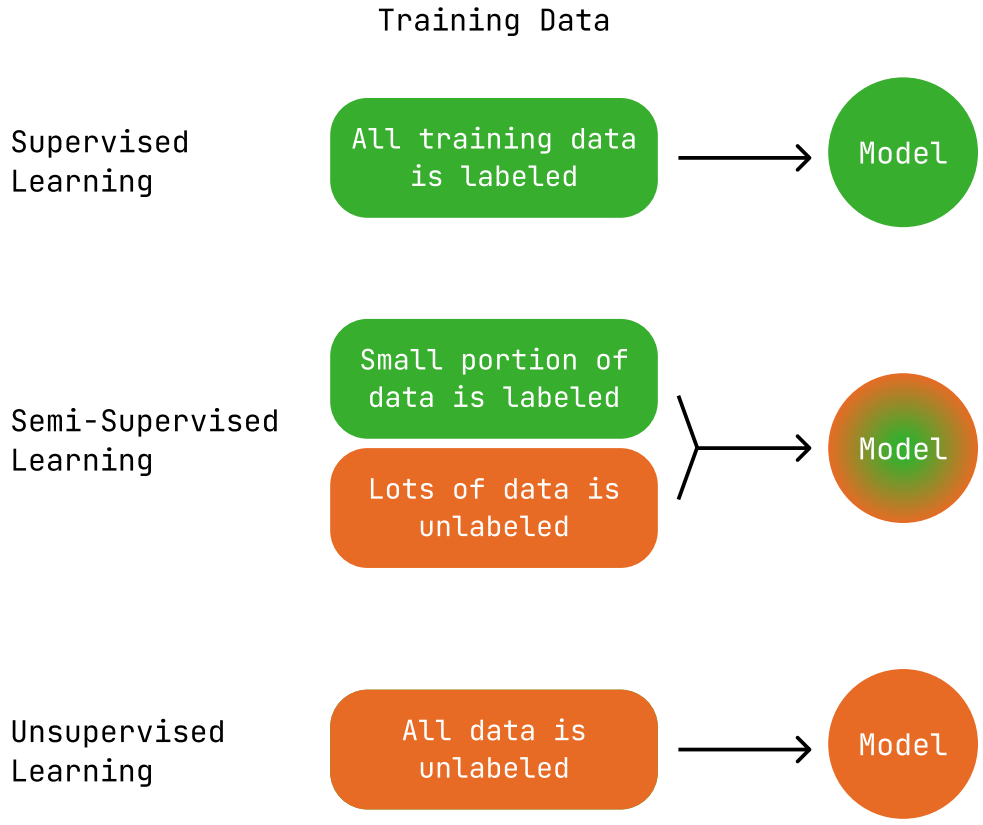

Figure I.3: Supervised Learning vs Semi-Supervised Learning vs Unsupervised Learning [41]

3.2.3 Hybrid Approach

A hybrid approach may be defined as a combination of lexicon-based approaches and machine learning [3]. This often results in improved classification accuracy.

3.2.4 Deep Learning Approach

Deep learning (DL) is a subset of machine learning (ML) where the learning occurs through multilayered neural networks (referred to as deep neural networks) with the objective of simulating the complex decision-making capabilities of the human brain.
In the context of text classification, two of the most commonly used neural network architectures are convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [51]. This latter represents the base for LSTM.

Convolutional Neural Network (CNN) Neural networks are computational models inspired by the human brain, designed to recognize patterns and relationships within data. They encompass a diverse array of architectures, each tailored to specific types of data and applications. Among these, Convolutional Neural Networks (CNNs) are predominantly recognized for their efficacy in processing visual data, such as images and videos. However, their utility extends beyond visual data, as CNNs can also be adeptly employed for the analysis of textual data.

Long-Short Term Memory (LSTM) LSTM is a deep learning model that uses Artificial Neural Networks (ANN) to learn patterns and more specifically it is an improved version of RNN (Recurrent Neural Network) due to its ability to overcome the vanishing gradient problem caused by RNN.

3.2.5 Transfer Learning Approach

The transfer learning approach is based on the principle of reusing the knowledge acquired by a model during its training in task ($\mathscr{A}$) and applying it to task ($\mathscr{B}$), which has limited available data. This would eliminate our need for the significant computing and time resources, as well as the millions of labeled data points when building complex models. For example, regarding the NLP field, we can fine-tune a language pre-trained model on general language comprehension task to perform new task like sentiment analysis.

DistilBERT DistilBERT [40] is a transformer model based on the BERT model released by Google [14]. It is a distilled form of the BERT model resulting in 40% fewer parameters and 60% faster while maintaining over 95% of BERT's performance as measured by the GLUE language comprehension benchmark (refer to figure I.4).

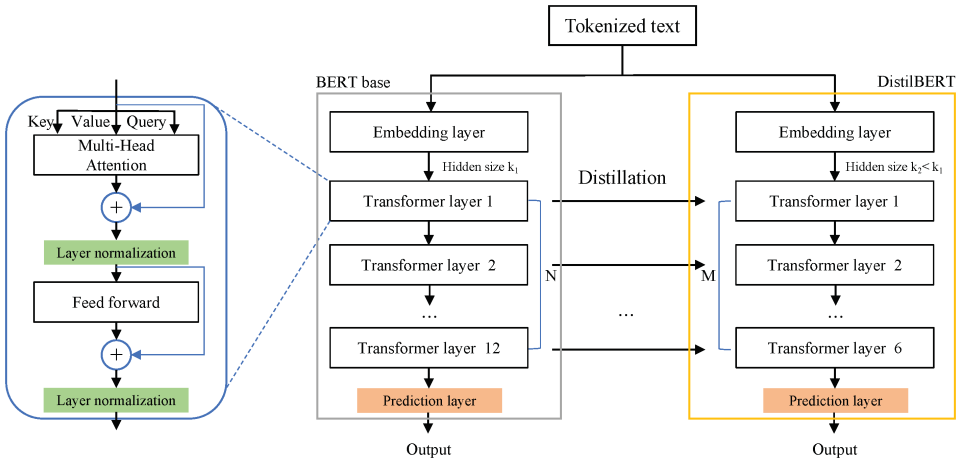

Figure I.4: The DistilBERT model architecture and components [2]

3.2.6 Some Advantages and Disadvantages of Different Approaches

Approach	Advantages	Disadvantages
Lexicon	Simple to use and does not require labeled data	Difficulty in understanding the sarcasm
Machine Learning	Ability to generalize well in the case of sufficient data of high quality	Costs are higher compared to the lexicon-based approach
Hybrid	Combines the strengths of lexicon and machine learning methods	It is more complicated and the cost can be higher
Deep Learning	Ability to capture complex patterns	Requires large amounts of data and high computing resources
Transfer Learning	Requires less labeled data and can achieve high accuracy scores	Fine-tuning can become complex

Table I.2: Some Advantages and Disadvantages of Different Approaches

3.3 Performance Evaluation

In their book, Vajjala et al. [51] identify accuracy, precision, recall, and F1 score as a widely recognized performance evaluation metrics for classification tasks.

3.3.1 Accuracy

It is the most common metric, and represents the answer to the question: ***"Of all the predictions we made, how many were correct?"***

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

3.3.2 Precision

Precision measures the proportion of true positive predictions among all positive predictions made by the model. It answers the question: ***"Out of all the positive predictions we made, how many were true?"***

$$Precision = \frac{TP}{TP + FP}$$

3.3.3 Recall (Sensitivity)

Also known as true positive rate, it focuses on how good the model is at finding all the positives. It answers the question ***"Out of all the data points that should be predicted as true, how many did we correctly predict as true?"***

$$Recall = \frac{TP}{TP + FN}$$

3.3.4 F1 Score

As mentioned earlier, the accuracy metric measures the number of correct predictions across the entire dataset. However, in real-world scenarios, and particularly in our case, the dataset is often imbalanced, making accuracy an unreliable metric.
To address this issue, we use the F1 Score, which calculates the harmonic mean of precision and recall. This metric is especially useful for evaluating performance on imbalanced datasets, as it provides a balanced measure that considers both false positives and false negatives.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4 Related Works

As previously stated, a multitude of data types can be employed for sentiment analysis. However, we will focus our research on textual data only.

A comprehensive study conducted by Al-Shabi [43] presents an in-depth comparison of various lexicon-based classifiers. The study utilized two distinct test sets: the Stanford Twitter Sentiment Test Set and the Sanders Twitter Test Set. The classifiers evaluated included

VADER, SentiWordNet, SentiStrength, AFINN-111, and Liu-Hu. The findings revealed that VADER outperformed the other classifiers, achieving an accuracy of 72% on the Stanford Twitter Sentiment Test Set and 65% on the Sanders Twitter Test Set, respectively.

In this study [12], researchers analyzed tweets from Indian citizens regarding the coronavirus crisis. The data was obtained from GitHub and annotated into four categories: fear, sad, anger and joy. The BERT model was fine-tuned and used for emotion recognition purposes. Its efficiency was evaluated by comparing it to three other models – LR, SVM and LSTM. The results indicated that the BERT model demonstrated an accuracy rate of 89%, which was significantly higher than the 75% achieved by its closest competitor, LR. Although the result was favorable, it could have been enhanced through additional preprocessing of the data prior to classification.

Similarly, in [52], the researchers aimed to analyze public sentiment towards the coronavirus vaccine in the Philippines. To this end, they collected data from Twitter, comprising 11,974 tweets from users in the country. These were manually annotated into three categories: positive, negative, and neutral. The researchers then employed the NB model to classify the tweets, after extracting features using TF-IDF. The 10-fold cross-validation technique was employed to assess the efficacy of the model, resulting in an accuracy score of 81.77%.

A research study was conducted by Muhammad, Kusumaningrum, and Wibowo [30] to analyze the sentiments expressed in Indonesian hotel reviews using Word2Vec and LSTM. The dataset comprised 2,500 texts, obtained from the Traveloka website, which were divided into two categories: positive and negative. Following the preprocessing stage, the Word2Vec algorithm was trained on the dataset in order to learn vector representations of words, utilizing a range of techniques and sets of parameters. The LSTM was tested with a variety of parameters, and among all the combinations and experiments, the optimal configuration achieved an average accuracy of 85.96%.

In order to study the public sentiments towards the Russian-Ukrainian conflict, a research was conducted by Wadhwani et al. [53] and his collaborators, where 25,000 tweets were scraped. After preprocessing, only 11,250 tweets remained, which represent a huge loss in the number of original data ($loss > 50\%$). A labeling phase was then conducted using the popular tool TextBlob in order to annotate the tweets into three categories: positive, negative and neutral. Subsequently, three distinct feature extraction methodologies (TF-IDF, BoW, N-Gram) were employed in conjunction with a multitude of supervised machine learning algorithms, including (RF, LR, DT, SVM, XGB, GNB, ADA, KNN, ETC and SGD). The highest accuracy was attained by ETC (in conjunction with the BoW technique) at 84%.

In another study focusing on the same Russian-Ukrainian conflict [31], data was collected using PRAW and labeled with VADER. After preprocessing, several classifiers were trained and evaluated. The Multinomial Naive Bayes (MNB) classifier achieved the highest accuracy and F1 score, with 82.65% and 76.53%, respectively. After cross-validation, the accuracy of MNB increased to 85.4%.

A study was made for a large Swedish telecom company to analyze customer sentiment [8]. The dataset consisted of more than 168,000 emails. An annotation phase was carried out using the VADER tool together with a Swedish lexicon. A feature extraction step was then performed using TF-IDF, followed by a selection step using χ^2. The two variants of SVM (SVC and LinearSVC) were used for classification. The results show that the LinearSVC model was able to extract sentiment with a mean F1 score of 0.834 and a mean AUC of 0.896.

In analyzing the sentiment expressed in food reviews, Ahmed et al. [4] employed the dataset Amazon Fine Food Reviews. One of the defining characteristics of this dataset is a score that ranges from 1 to 5. This was used to classify the data into two classes, positive and negative. Subsequently, TF-IDF was employed to extract features from the textual data. The researchers indicated that among the numerous models they tested, three models exhibited an accuracy of greater than 80%: NB, LinearSVC, and LR. The LinearSVC model achieved the highest accuracy of 88.38%, slightly ahead of the LR model (87.38%). However, it is not always the case that the scores accurately reflect the polarity of the sentiments, which raises questions about the reliability of the annotation phase.

In the entertainment industry, study [56] was based on sentiment analysis of movie reviews. For this purpose, the popular IMDB dataset was utilized. After preprocessing, attribute selection was conducted using the gain ratio algorithm. A comparison was then made between eight different classification models, with RF achieving the highest accuracy (96.01%).

Another study [37] focused on IMDb reviews utilized an LSTM classifier for categorizing comments into positive and negative sentiments. Following segmentation and processing, Doc2Vec was employed for feature extraction. The LSTM architecture comprised three layers. Ultimately, this approach achieved a classification accuracy of 89.9%.

Similarly, in the study conducted by Pipalia, Bhadja, and Shukla [35], the same dataset was employed, but different approaches were utilized. Specifically, the research compared the performance of a bidirectional LSTM with five transformer-based models, namely BERT-Base, RoBERTa, XLNet, T5, and DistilBERT. The results indicated that XLNet outperformed all other models, achieving an accuracy of 96.2%. In contrast, the bidirectional LSTM demonstrated the lowest performance among the models tested, with an accuracy of 86.6%.

Study	Dataset	Labeling	Classifiers	Best Result
[43]	Stanford Twitter & Sandars Twitter (Test Sets)	(positive, negative and neutral)	VADER,enti-WordNet, Sen-tiStrength, AFINN-111, and Liu-Hu	VADER (ACC): 72% and 65%
[12]	Tweets (From GitHub)	Manually (fear, sad, anger, joy)	LR, SVM, LSTM and BERT	BERT (ACC): 89%
[52]	Gathered Tweets	Manually (positive, negative and neutral)	NB	NB (ACC): 81.77%
[30]	Crawled from Traveloka website	Positive - Negative	LSTM	LSTM (ACC): 85.96%
[53]	Scraped from Twitter	TextBlob (positive, negative and neutral)	RF, LR, DT, SVM, XGB, GNB, ADA, KNN, ETC and SGD	ETC (ACC): 84%
[31]	Scraped from Reddit using PRAW	VADER (positive, negative and neutral)	MNB, BernoulliNB, GNB, and RF	MNB (ACC): 85.4%
[8]	customer support e-mails (Swedish telecom company)	VADER + Swedish lexicon (very negative, negative, neutral, positive, very positive)	LinearSVC, SVC	LinearSVC (F1): 83.4%
[4]	Amazon Fine Food Reviews	Positive - Negative	NB, LR, LinearSVC, and others	LinearSVC (ACC): 88.38%
[56]	IMDB reviews dataset	Positive - Negative	NB, DT, SVM, BN, KNN, RRL, RF, SGD	RF (ACC): 96.01%

Study	Dataset	Labeling	Classifiers	Best Result
[37]	IMDB reviews Dataset	Positive - Negative	LSTM	LSTM (ACC): 89.9%
[35]	IMDB reviews Dataset	Positive - Negative	Bi-LSTM, BERT-Base, RoBERTa, XLNet, T5, and DistilBERT	XLNet (ACC): 96.2%

Table I.3: Taxonomy

5 Conclusion

As we have seen above, the field of sentiment analysis is extremely important and widely used. Investing efforts and research in it will benefit many people and entities.
In this chapter, we have reviewed some conceptual basics in the field of sentiment analysis. In the next chapter we will discuss our work methodology in detail.

CHAPTER II

CONCEPTION

Contents

1	**Introduction**	19
2	**General Architecture**	19
3	**Methodology**	20
4	**Conclusion**	26

1 Introduction

As previously discussed, the applications of sentiment analysis extend across various domains. One prominent domain where SA is employed is politics. This research addresses a political conflict, specifically the Palestinian-Israeli conflict and the recent war on Gaza. For years, the Zionist narrative has dominated the media, portraying its actions as self-defense against so-called terrorists. However, the recent war has unveiled new perspectives, prompting global discourse and differing opinions.

Given these considerations, we aim to conduct a comparative analysis of the techniques used for sentiment analysis concerning this conflict. In the second chapter, we will present our methodology in detail, providing a comprehensive explanation of our approach.

2 General Architecture

Our architecture, as illustrated in Figure II.1, outlines the key stages of our approach, consisting of seven essential steps.

The process begins with data collection, a foundational phase where relevant data is gathered.

This is followed by a labeling phase, where the collected data is annotated to facilitate supervised learning. Subsequently, a preprocessing step is undertaken to clean and prepare the data for analysis. Next, feature extraction is performed to transform the text data into a numerical format suitable for machine learning algorithms. Following that, classifiers are constructed to categorize the data. These classifiers are then evaluated to ensure their effectiveness. Finally, the validated classifiers are deployed, completing the workflow and enabling real-world application of the developed models.

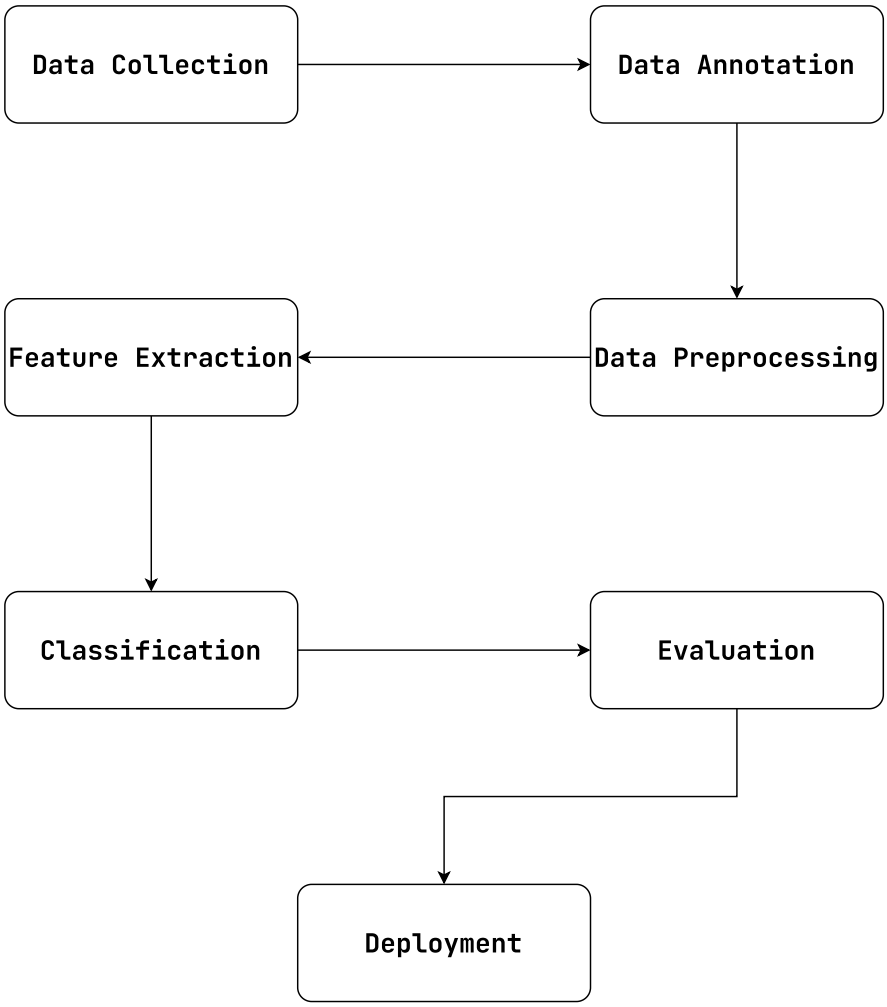

Figure II.1: General Architecture

3 Methodology

3.1 Data Collection

The initial phase of our research entails the collection of data, with a particular focus on public comments or posts pertaining to the war in Gaza. While numerous data sources are available,

including social media sites, news websites, blogs, and surveys, we have chosen to focus exclusively on social media platforms. This decision is based on the high level of user engagement and the rich, real-time discourse that these platforms provide.

There are several approaches to data collection. One option is to utilize a pre-existing dataset, which, in this context, is often scarce due to the recency of the conflict. Another is to manually gather data, which is impractical given the requirement for tens of thousands, if not millions, of individual opinions. A third option is to employ APIs offered by social media platforms and web scraping techniques. These methods must be carefully considered in light of the legal constraints imposed by the respective companies and the permissibility of such actions.

Given the nascent stage of the current war and the consequent paucity of available datasets, our methodology will rely on accessing data through APIs or employing web scraping techniques.

Therefore, we chosen utilizing the API provided by the popular social news and discussion site Reddit, While it is still not as widely used as Facebook or X (formerly Twitter), it offers a way for users to share news without being overly restrictive. This is in contrast to Meta, which has faced criticism recently, particularly regarding the restriction of pro-Palestinian content under the pretext of anti-Semitism.

With respect to X (formerly known as Twitter), our decision to exclude this platform, despite its extensive discourse on the ongoing conflict, is informed by the recent policy changes implemented by its current owner, Elon Musk. Specifically, Musk's decision to terminate the free API and impose prohibitively high fees for queries has rendered the platform impractical for our research purposes [6]. Consequently, we have identified Reddit as a more viable and cost-effective alternative for data collection and analysis under the present circumstances.

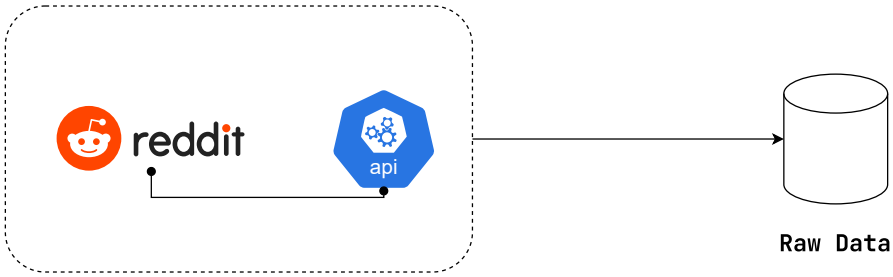

Figure II.2: Data Collection

3.2 Data Annotation (VADER)

In order to annotate the collected data, we used the Valence Aware Dictionary for sEntiment Reasoning (VADER) [20] which is a highly accurate lexicon and rule-based tool for opinion mining, particularly effective in analyzing social media text. It relies on a dictionary of words and predefined rules to assess sentiment. Each word is assigned a valence score, indicating its polarity, with values ranging from -4 to 4 for negative and positive sentiments, respectively.

VADER also considers the intensity of sentiment, which can be inferred from factors like capitalization and punctuation (capital letters or exclamation marks may suggest a stronger sentiment).

3.3 Data Preprocessing

Considering that Reddit's raw comments are usually long, unstructured, noisy, and often contain HTML code, links, emoticons and emojis, we should preprocess them through the following steps:

- **Deleted Comments:** After extracting the data, the comments that the user has previously deleted will appear as **[removed]**. There is no need to keep them, so we should delete them.
- **Links:** It might be worth considering removing links from social media posts and comments, as they don't necessarily add anything to the sentiment analysis process.
- **Removing HTML Tags, mentions (@) and Subreddits(r/).**
- **Contractions:** It is important to make sure that the original word and its contraction [1] should be considered as the same word, so a transformation to the long form is needed.
- **Slangs and Abbreviations:** We tried to convert some common slang [2] words and acronyms back to their full forms.
- **Lowercasing:** It involves converting all letters to lowercase to ensure uniformity and consistency of words, regardless of their original capitalization.
- **Tokenization:** The process of tokenization involves the division of text into smaller units, each one of them called token.
- **Removing special characters, emojis and emoticons.**
- **Part-of-Speech Tagging (POS) and Lemmatization:** POS is common grammatical NLP technique represent the process of categorizing words in a text to a nouns, verbs, adjectives ..etc. It was employed alongside with the lemmatization technique which describe the process of taking a word and breaks it down to its lemma.

In addition to the aforementioned steps, an additional procedure was employed specifically for deep learning and transfer learning models, which is:

[1] Contractions represents a short form of a word or combination of words that is often used instead of the full form in spoken English [13].

[2] Slang refer to a very informal language that is used especially in speech by particular groups of people and which sometimes includes words that are not polite [45].

- **Padding:** Padding in text processing refers to the practice of appending zeros or designated padding tokens to shorter sequences, with the objective of aligning them with the length of the longest sequence in the dataset. It ensures that all input sequences have a uniform length, thereby facilitating efficient batch processing and training of NN.

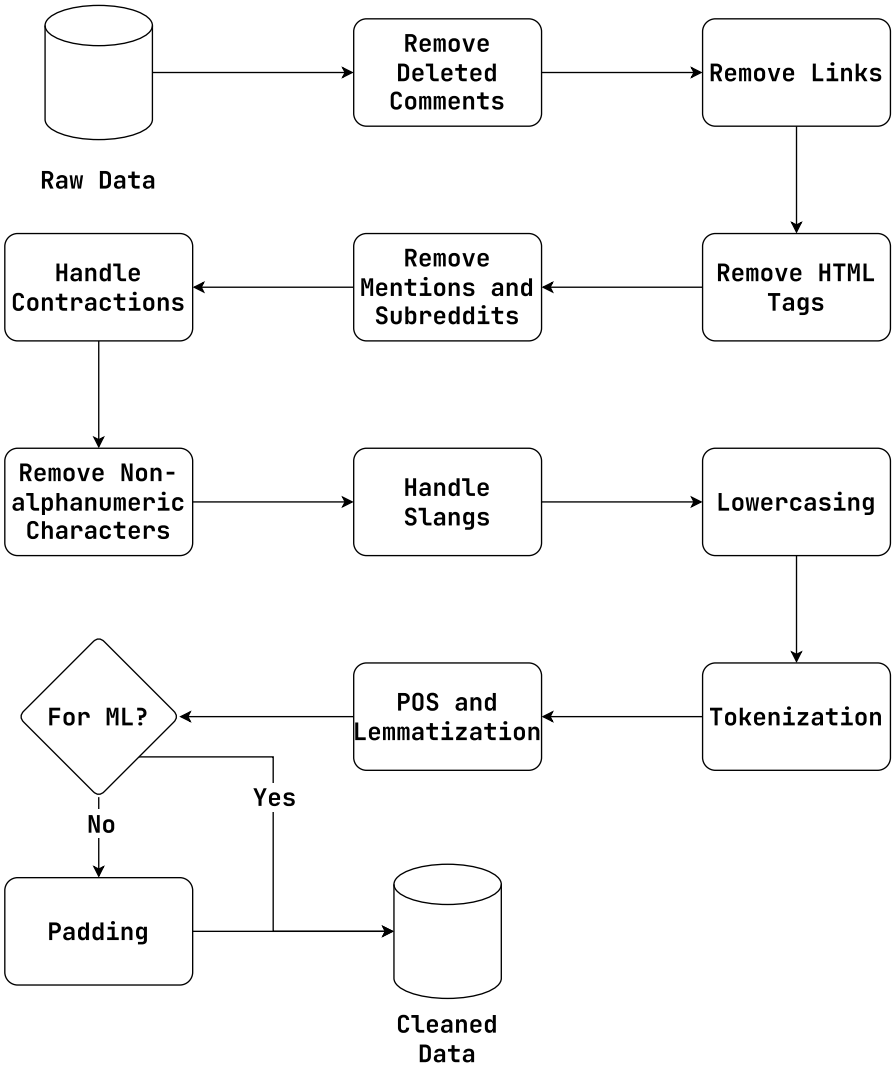

Figure II.3: Data Preprocessing

3.4 Feature Extraction

Feature extraction has a crucial role in text classification systems. It aims to transform an input text document into a numerical representation that can be processed by a machine.
In our approach, we used vectorization methods for traditional machine learning classifiers, including Term Frequency – Inverse Term Frequency (TF-IDF) and Bag of Words (BoW).
Furthermore, we utilized the pre-trained Word embedding and the pre-trained model, GloVe and Google-News Word2Vec respectively, for deep learning classifiers.

3.5 Classification

To conduct a comprehensive study that incorporates various types and approaches of sentiment analysis (SA), we employed four different machine learning-based classifiers: Multinomial Naive Bayes (MNB), Logistic Regression (LR), Linear Support Vector Classifier (LinearSVC), and Decision Tree (DT). Additionally, we utilized two deep learning-based classifiers, Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN).
The LSTM model architecture consists of three LSTM layers, each designed to capture sequential dependencies in the text data, ending in a dense layer with a softmax activation function to perform the final classification (refer to Figure II.4). On the other hand, the CNN model architecture comprises three convolutional layers with decreasing filter sizes, each followed by a max-pooling layer to down-sample the feature maps, and concludes with a dense layer with a softmax activation function for classification (refer to Figure II.5) .
To ensure completeness and enhance our analysis, we also fine-tuned the DistilBERT transformer-based model on our data to perform sentiment classification. This diverse ensemble of models allows us to thoroughly explore and compare the effectiveness of traditional machine learning techniques against advanced deep learning and transformer-based approaches in sentiment analysis.

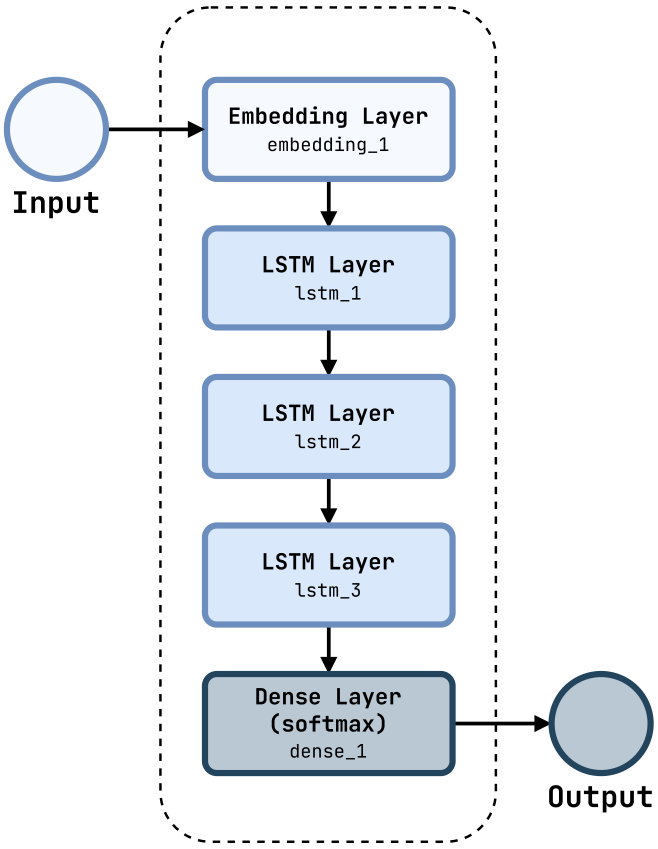

Figure II.4: Architecture of LSTM Model

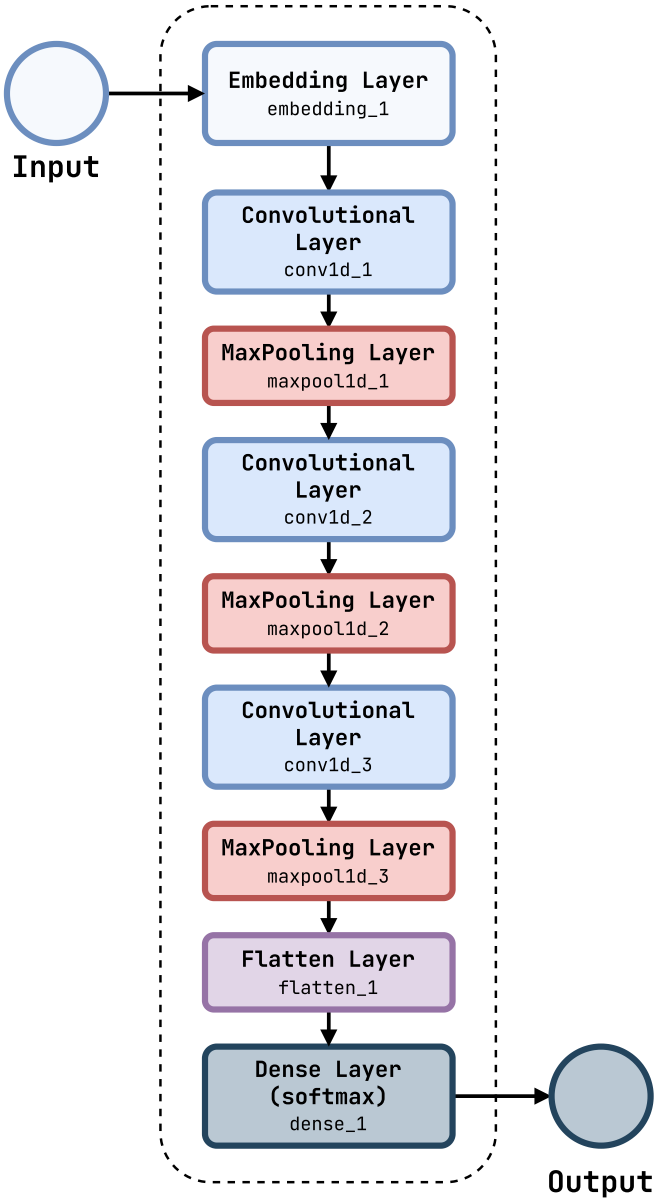

Figure II.5: Architecture of CNN Model

3.6 Evaluation

To evaluate the performance of our models, we employed four widely recognized metrics: accuracy, precision, recall, and F1-score (refer to the subsection 3.3 from the first chapter).

3.7 Deployment

After evaluating our seven models, we selected the best-performing ones for production. Those models were then integrated into a website, allowing users to input custom text and choose

the desired model to obtain sentiment analysis results.

4 Conclusion

In this chapter, we describe in detail our methodology for analyzing public opinion on the Palestinian-Israeli conflict. This explanation includes all steps from data collection to the evaluation of the classifiers' performance.
In the next chapter, we will address the practical application of the methodology and discuss the results we obtained.

CHAPTER III

IMPLEMENTATION AND RESULTS

Contents

1	**Introduction**	27
2	**Work Environment**	28
3	**Programming Language and Libraries**	28
4	**Implementation**	29
5	**Obtained Results**	38
6	**Additional Experiments**	45
7	**Conclusion**	50

1 Introduction

Building upon the comprehensive methodology presented in the preceding chapter, this chapter focuses on the practical aspects of our research. We commence by delineating the operational environment, specifying the programming language employed, and enumerating the various libraries and tools that are integral to our approach. Subsequently, illustrative code snippets are provided to elucidate key elements of the implementation. Subsequently, the results obtained are presented, accompanied by a thorough analysis of the outcomes of various experiments conducted. This comprehensive exposition is intended to provide a clear and precise understanding of the practical execution and the empirical findings of our study.

2 Work Environment

2.1 Google Colab

Google Colab or **Colaboratory** is a hosted Jupyter Notebook that provides free computing resources, including GPUs. It is compatible with Google services such as Google Drive, which makes it an ideal platform for machine learning and data science in general. In our case, We used it to gather, store, annotate, and preprocess the data.

2.2 Kaggle

Kaggle is an online community platform for data science competitions. Participants compete to create the most effective models for solving specific problems or analyzing certain datasets. The platform is also used for learning, collaboration, and research in the data science and machine learning fields. It hosts a Jupyter Notebook, similar to G-Colab, in addition to a wide range of datasets. The platform was selected for use during the training phase due to its flexibility in affording the free GPUs or TPUs. The free version allows for 30 hours of GPU usage per week, with 20 hours permitted for TPUs.

3 Programming Language and Libraries

3.1 Python

Python is an interpreted, object-oriented, high-level and open-source programming language that has achieved widespread adoption and acclaim on a global scale. Python is renowned for its user-friendly syntax and ease of learning, which have contributed to its status as one of the most popular and versatile languages in the programming community. Its simplicity and readability render it an optimal choice for beginners, while its robust libraries and frameworks appeal to experienced developers across a range of domains including web development, data science, artificial intelligence, and machine learning. This reflects the language's extensive functionality and adaptability.

3.2 Libraries

Python offers a robust collection of powerful libraries in the machine learning field, and we employed several of these in our implementation:

3.2.1 Python Reddit API Wrapper

Python Reddit API Wrapper (PRAW) is a Python package that facilitates interaction with the Reddit API. It enables users to retrieve posts, comments, and user's public information, as

well as to post and moderate content. In our case we used it to collect comments.

3.2.2 Pandas

Pandas [50] is a highly popular and widely used tool by data scientists. It simplifies the way we interact with datasets (particularly big data). It offers many functions that help in analyzing, cleaning, exploring and manipulating data. we used it nearly in every phase of our work.

3.2.3 Natural Language ToolKit (NLTK)

Natural Language Toolkit (NLTK) [7] is a Python programming environment for building applications for natural language processing (NLP). It contains language processing libraries for tokenization, stemming, lemmatization, sentiment classification, among others.

3.2.4 Scikit-Learn

Scikit-learn (sklearn for short) [33] is a highly useful and robust Python library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling, including classification, regression, clustering, and dimensionality reduction. It was employed during the splitting phase, feature extraction phase and modeling phase.

3.2.5 TensorFlow

TensorFlow [1] is an open-source library developed by Google that is specifically designed to create deep learning applications. Renowned for its extensive pre-built functions, TensorFlow provides a straightforward and efficient means of implementing complex neural network architectures.

3.2.6 PyTorch

PyTorch [32] is another open source fully featured framework for building deep learning models based on the Python programming language and the Torch library. PyTorch is renowned for its competitive relationship with TensorFlow in the field.

4 Implementation

4.1 Data Collection

The data was obtained by collecting comments from various trending subreddits (**r/IsraelPalestine**, **r/Palestine**, **r/worldnews**, **r/AskMiddleEast**, **r/Israel**, and **r/CombatFootage**) using the

PRAW library. All of these comments were posted after October 7, 2023, ensuring that our analysis targets people's opinions on the latest conflict only.

```python
import praw
# Initialize the Reddit instance (the credentials are configured during the registration process to the Reddit API.)
reddit = praw.Reddit(client_id=CLIENT_ID, client_secret=SECRET_KEY, user_agent=USER_AGENT)

# A list to store comments
comments = []
# Specifying the subreddit to scrape from
subreddit = "IsraelPalestine"

# Loop through the top 100 submissions in the specified subreddit
for submission in reddit.subreddit(subreddit).top(limit=100):
    # Loop through the comments of each submission (post)
    for comment in submission.comments:
        if isinstance(comment, praw.models.MoreComments):
            continue
        # Append the comment body to the comments list
        comments.append(comment.body)

        # Additional information that can be retrieved:
        # score: comment.score
        # author name: comment.author.name
        # ...
```

Code Snippet III.1: Data Collection (PRAW)

To ascertain the language distribution within this dataset, we employed the Python package `langdetect`, which indicates that a substantial majority of the comments (92.96%) were in English. Consequently, we have narrowed the focus of our study to this subset of English-language comments in order to ensure consistency in our analysis.

4.2 Data Annotation

Labels were assigned to each document following the procedure detailed in Code Snippet III.2. This implementation utilized the NLTK library. Regarding the threshold values of 0.05 and -0.05, it is important to note that these are commonly used values and not fixed constants.

```python
from nltk.sentiment.vader import SentimentIntensityAnalyzer

analyzer = SentimentIntensityAnalyzer()

def get_sentiment(text):
    scores = analyzer.polarity_scores(text)
    if scores['compound'] >= 0.05:
        sentiment = 'Positive'
    elif scores['compound'] <= -0.05:
        sentiment = 'Negative'
    else:
        sentiment = 'Neutral'
    return sentiment
```

Code Snippet III.2: Data Annotation (VADER)

4.3 Data Preprocessing

The preprocessing phase, as detailed in subsection 3.3 of the second chapter, comprised several steps. The implementation involved the use of regular expressions to replace and remove specific patterns, the application of BeautifulSoup to eliminate HTML tags when present, and the utilization of a custom slang dictionary to address abbreviations. Additionally, the NLTK library was employed for tokenization and lemmatization, with the corresponding code illustrated in Code Snippet III.3.

```python
# Example of slangs dictionary
slang_dict = {"u": "you", "btw": "by the way", "b4": "before"}

def preprocessing(comment):
    # Removing links and URLs
    no_links = re.sub(r'https?:\S*', '', comment)
    # Removing HTML tags
    no_html = BeautifulSoup(no_links, 'html.parser').get_text()
    # Removing mentions and subreddits
    no_mentions = re.sub(r'@\S*', '', no_html)
    no_subreddits = re.sub(r'r/\S*', '', no_mentions)
    # Replacing contractions
    no_contractions = contractions.fix(no_subreddits)
    # Removing all special characters and emojis
```

```
    no_sc = re.sub(r'[^\w\s]', '', no_contractions)
    # Replacing slangs
    no_slangs = [slang_dict[word] if word in slang_dict else word for
     ↪ word in no_sc.split(' ')]
    # Tokenization
    tokens = word_tokenize((' '.join(no_slangs)).lower())
    # Part-of-speech tagging
    pos_tags = nltk.pos_tag(tokens)
    # Lemmatization based on POS tags
    lemmatized_tokens = []
    for token, pos_tag in pos_tags:
        if pos_tag.startswith('J'):
            wordnet_pos = 'a'  # Adjective
        elif pos_tag.startswith('V'):
            wordnet_pos = 'v'  # Verb
        elif pos_tag.startswith('N'):
            wordnet_pos = 'n'  # Noun
        elif pos_tag.startswith('RB'):
            wordnet_pos = 'r'  # Adverb
        else:
            wordnet_pos = 'n'  # Default to noun
        # Lemmatize the token
        lemmatized_token = lemmatizer.lemmatize(token, pos=wordnet_pos)
        lemmatized_tokens.append(lemmatized_token)
    # Removing non-alphabetic tokens (e.g, numbers)
    clean_tokens = [token.lower() for token in lemmatized_tokens if
     ↪ token.isalpha()]
    return ' '.join(clean_tokens)
```

Code Snippet III.3: Data Preprocessing

However, when utilizing non-ML classifiers, we perform padding as detailed in subsection 3.3. The maximum number of tokens in our corpus is 160 tokens; therefore, we padded the text to this length. This process is illustrated in Code Snippet III.4.

```
from tensorflow.keras.preprocessing.sequence import
 ↪ pad_sequences
pad_train = pad_sequences(tokenizer.texts_to_sequences(texts), maxlen=160)
```

Code Snippet III.4: Padding

4.4 Data Set Splitting

Following the data pre-processing phase, the dataset was partitioned into two subsets: 80% for training and 20% for testing. The class distribution was preserved across these subsets through the use of the `stratify` parameter (refer to the Code Snippet III.5), ensuring an even representation of classes in both the training and testing sets.

```
from sklearn.model_selection import train_test_split

X = dataset["comment"]
y = dataset["label"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
 ↪  random_state=1, stratify=y)
```

Code Snippet III.5: Data Splitting

4.5 Feature Extraction

As previously stated in subsection 3.4, we employed a number of feature extraction techniques, including Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), the pre-trained GloVe embeddings, and the pre-trained Google-News Word2Vec model. These techniques, while sharing a common objective of representing textual data numerically, differ significantly in their implementation.
For BoW and TF-IDF, we utilized the straightforward implementations available in the scikit-learn library (Code Snippet III.6)

```
from sklearn.feature_extraction.text import TfidfVectorizer,
 ↪  CountVectorizer

# for BoW, we can use CountVectorizer()
vectorizer = TfidfVectorizer()
# fit the vectorizer and transform the train data
features_train = vectorizer.fit_transform(X_train)
# use the fitted vectorizer to transform the test data
features_test = vectorizer.transform(X_test)
```

Code Snippet III.6: TF-IDF and BoW

In contrast, working with GloVe embeddings requires handling a text file containing pre-trained vectors. We employed a custom script to load this file and extract the vector representation for each word. The pre-trained embeddings are available in various dimensionalities,

including 50, 100, 200, and 300 dimensions. For this study, we opted to use the 300-dimensional embeddings to capture a more comprehensive range of semantic meanings.

```
embeddings = {}
with open('glove.6B.300d.txt', encoding='utf-8') as f:
    for line in f:
        values = line.split()
        word = values[0]
        embeddings[word] = np.asarray(values[1:], dtype='float32')
```

Code Snippet III.7: GloVe

For the Word2Vec model, we opted to use the `gensim` library, which simplifies the handling of the binary model file. `gensim` provides a high-level interface for loading and querying pre-trained Word2Vec models, allowing us to efficiently obtain word vectors and perform various vector space operations.

```
from gensim.models import KeyedVectors

word_vectors = KeyedVectors.load_word2vec_format(
    "GoogleNews-vectors-negative300.bin", binary=True)
```

Code Snippet III.8: Word2Vec

Following the acquisition of pre-trained word embeddings, an embedding matrix is constructed for initializing the embedding layer (refer to Code Snippet III.9).

```
from tensorflow.keras.initializers import Constant
from tensorflow.keras import layers

embedding_matrix = np.zeros((vocab_size, 300))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
embedding_layer = layers.Embedding(input_dim=vocab_size, output_dim=300,
 ↪  embeddings_initializer=Constant(embedding_matrix), trainable=False,
 ↪  mask_zero=True)
```

Code Snippet III.9: Embedding Matrix

4.6 Classification

In the classification phase, a diverse set of classifiers was employed, encompassing machine learning, deep learning, and transfer learning paradigms.
For the machine learning classifiers, four specific models were implemented: MNB, LinearSVC, LR, and DT. These models were developed using the scikit-learn library, which facilitated an efficient and straightforward implementation process (refer to Code Snippet III.10).

```
from sklearn.naive_bayes import MultinomialNB
# from sklearn.svm import LinearSVC
# from sklearn.linear_model import LogisticRegression
# from sklearn.tree import DecisionTreeClassifier

# Initialize the classifier (e.g, MNB)
clf = MultinomialNB()
# Train the classifier on the train data
clf.fit(features_train, y_train)
```

Code Snippet III.10: Training a ML-Based Classifier (e.g, MNB)

In the deep learning classification segment, we utilized Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN). The architectural details of these models are elaborated in subsection 3.5. We configured the necessary layers for both models using the TensorFlow library. Detailed implementation examples for LSTM and CNN can be found in Code Snippets III.11 and III.12, respectively.

```
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential

model = Sequential([
    embedding_layer,
    layers.SpatialDropout1D(0.6),
    layers.LSTM(300, dropout=0.4, recurrent_dropout=0.4,
    ↪ return_sequences=True),
    layers.LSTM(128, dropout=0.2, recurrent_dropout=0.2,
    ↪ return_sequences=True),
    layers.LSTM(64, dropout=0.2, recurrent_dropout=0.2),
    layers.Dense(3, activation='softmax')
])
```

```
13
14 model.compile(loss='sparse_categorical_crossentropy',
15 optimizer=tf.keras.optimizers.Adam(learning_rate=8e-4),
    ↪ metrics=['accuracy'])
```

Code Snippet III.11: Initializing the LSTM Model

```
1  import tensorflow as tf
2  from tensorflow.keras import layers
3  from tensorflow.keras.models import Sequential
4
5  model = Sequential([
6      embedding_layer,
7      layers.Dropout(0.6),
8      layers.Conv1D(300, 4, padding='same', activation='relu'),
9      layers.MaxPool1D(1),
10     layers.Conv1D(128, 4, padding='same', activation='relu'),
11     layers.MaxPool1D(1),
12     layers.Conv1D(64, 4, padding='same', activation='relu'),
13     layers.MaxPool1D(1),
14     layers.Flatten(),
15     layers.Dense(3, activation='softmax')
16 ])
17
18 model.compile(loss='sparse_categorical_crossentropy',
19 optimizer=tf.keras.optimizers.Adam(learning_rate=5e-4),
    ↪ metrics=['accuracy'])
```

Code Snippet III.12: Initializing the CNN Model

However, it is essential to acknowledge that we encoded the labels before start the training (see Code Snippet III.13).

```
1 from sklearn.preprocessing import LabelEncoder
2 # Encode the labels
3 label_encoder = LabelEncoder()
4 y_train_encoded = label_encoder.fit_transform(y_train)
5 y_val_encoded = label_encoder.transform(y_val)
```

```
# Fit the model
model.fit(pad_train, y_train_encoded, validation_data=(pad_val,
 ↪  y_val_encoded), epochs=EPOCHS, batch_size=BATCH_SIZE)
```

Code Snippet III.13: Training a DL-Based Classifier

For the transfer learning approach, we fine-tuned the DistilBERT model [40] (`distilbert-base-uncased`) from Hugging Face [15] on our dataset. The fine-tuning process involved adapting the pre-trained model to our specific dataset requirements. We closely followed the comprehensive documentation provided by Hugging Face to ensure the accuracy and efficiency of the model fine-tuning process [49].

4.7 Evaluation

For the evaluation and implementation of the metrics, we utilized the scikit-learn library.

```
from sklearn.metrics import accuracy_score, f1_score,
 ↪  precision_score, recall_score

y_pred = clf.predict(features_test)

accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred, average='macro')
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')
```

Code Snippet III.14: Performance Evaluation

4.8 Deployment

To facilitate the deployment of the most effective models identified, we developed a web interface. This interface utilizes HTML and PicoCSS for the frontend, while the Flask Python framework is employed for the backend. Users can input their thoughts on the war in Gaza and select one of the seven models. After a brief processing period, they will receive the sentiment analysis results.

```
from flask import Flask, jsonify, render_template

# Initialize the Flask application
```

```
app = Flask(__name__)

# Define the route for the home page
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/get_sentiment', methods=['POST'])
def get_sentiment():
    # Get the JSON data from the POST request
    data = request.get_json()
    model = data['model']
    text = data['text']
    # Use the model to predict the sentiment of the text
    sentiment = model.predict(text)
    # Return the sentiment as a JSON response
    return jsonify({'sentiment': sentiment})

if __name__ == "__main__":
    app.run()
```

Code Snippet III.15: Flask - General Implementation

5 Obtained Results

5.1 Collected Data

The dataset utilized in this study comprises a total of 80,970 comments, meticulously curated from six trending subreddits, all of which are centered around discussions related to the war in Gaza. This substantial collection of data provides a diverse and representative sample of public opinion and sentiment across various online communities. Figure III.1 illustrates the distribution of comments across the different subreddits.

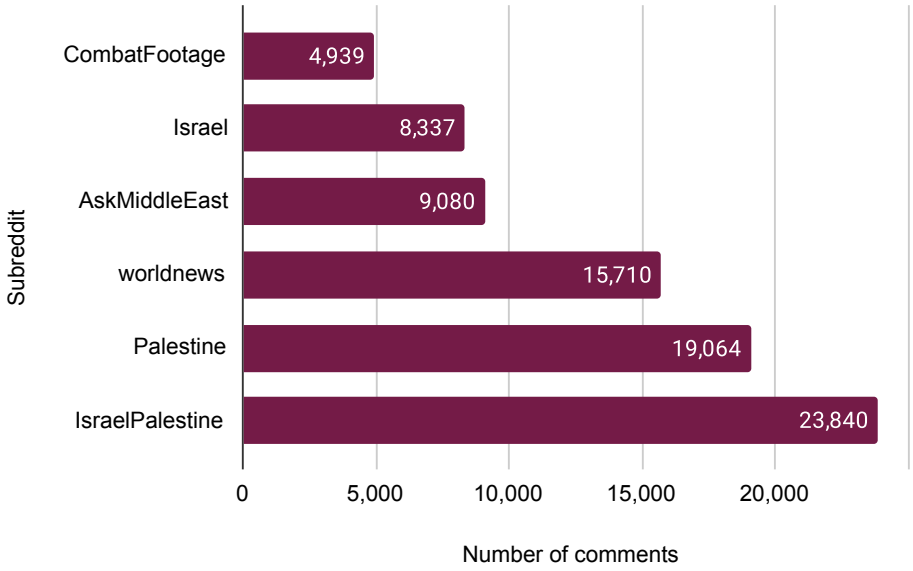

Figure III.1: Number of Comments in Each Subreddit

Figure III.2: Wordcloud of The Most Frequent Words

5.2 Labeled Data

As anticipated, the resulting sentiment distribution, illustrated in Figure III.3, demonstrates that the majority of comments express negative sentiments regarding the war. This finding aligns with our expectations based on the context and nature of the subject matter.
A sample of these comments is presented in Table III.1, along with their corresponding compound sentiment scores.

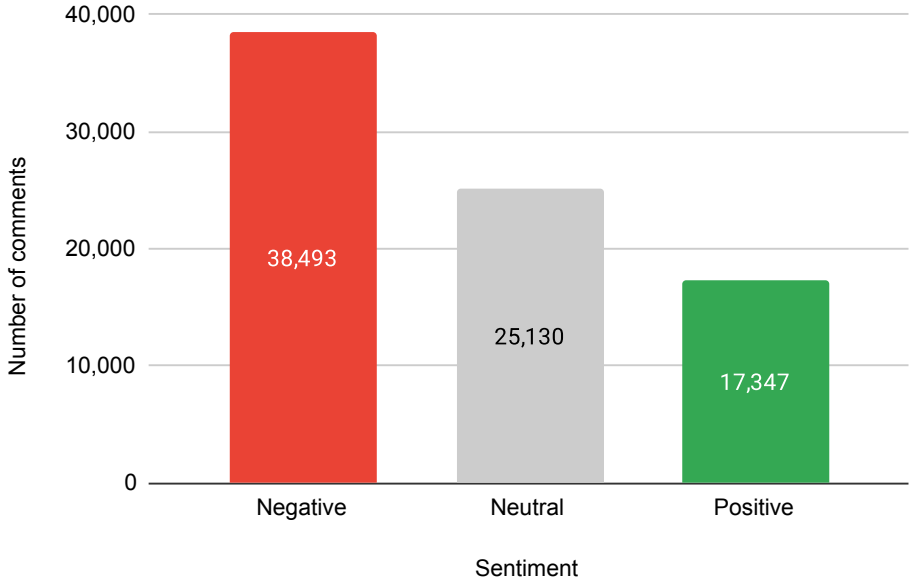

Figure III.3: Number of Comments by Sentiment

Comment	Compound	Label
Palestinians aren't only being brutalised n traumatised by the Zionists, but even their own so called "government" that the Israelis planted just to use it as an excuse to bomb these innocents to oblivion	-0.3241	Negative
Bibi will resign, but not because we think he wouldn't give us "peace" but because he made a disaster I hope bennet would be reelected	-0.5423	Negative
Didnt haaretz just come out with an article showing how the people slaughtered by israel were actually victims of the IDF? I think it diiiiiiiddddd	-0.3182	Negative
I applaud 👏 South Africa!!!!	0.6331	Positive
Poeple call them terrorists now but our grand children will read about Hamas as Freedom figthers fighting against an Apartheid state	0.7184	Positive
Beautiful to see <3	0.7783	Positive
This is what their parents raised them on.	0	Neutral
Zionazi is the correct term anyway	0	Neutral
Ask nethanyaho, ben gafir and smotritch about this, they have the answers.	0	Neutral

Table III.1: Sample of The Labeled Comments

5.3 Data After Preprocessing

As illustrated in Table III.2, the processing of comments yields results that are not entirely perfect. Several challenges contribute to this imperfection. Notably, the handling of slang terms remains inadequate, leading to potential misinterpretations. Additionally, the current methods struggle with repeated words and misspellings, further complicating the analysis. These limitations highlight the inherent difficulties in accurately processing and interpreting informal and unstructured text data.

Before Preprocessing	**After Preprocessing**
Palestinians aren't only being brutalised n traumatised by the Zionists, but even their own so called "government" that the Israelis planted just to use it as an excuse to bomb these innocents to oblivion	*palestinian be not only be brutalise n traumatise by the zionist but even their own so called government that the israeli plant just to use it a an excuse to bomb these innocent to oblivion*
Bibi will resign, but not because we think he wouldn't give us "peace" but because he made a disaster I hope bennet would be re-elected	*bibi will resign but not because we think he would not give u peace but because he make a disaster i hope bennet would be reelect*
Didnt haaretz just come out with an article showing how the people slaughtered by israel were actually victims of the IDF? I think it diiiiiiiddddd	*do not haaretz just come out with an article show how the people slaughter by israel be actually victim of the idf i think it diiiiiiiddddd*
I applaud 👏 South Africa!!!!	*i applaud south africa*
Poeple call them terrorists now but our grand children will read about Hamas as Freedom figthers fighting against an Apartheid state	*poeple call them terrorist now but our grand child will read about hamas a freedom figthers fight against an apartheid state*
Beautiful to see <3	*beautiful to see*
This is what their parents raised them on.	*this be what their parent raise them on*
Zionazi is the correct term anyway	*zionazi be the correct term anyway*
Ask nethanyaho, ben gafir and smotritch about this, they have the answers.	*ask nethanyaho ben gafir and smotritch about this they have the answer*

Table III.2: Before and After Preprocessing

5.4 Training Set and Testing Set

Following the preprocessing stage, the number of remaining comments was 70,268. The following Table III.3 presents the distribution of comments by category (negative, positive and neutral) between the training set and the testing set.

Sentiment	Training Set	Testing Set	Total
Negative	27,653	6,914	34,567
Positive	18,072	4,518	22,590
Neutral	10,489	2,622	13,111
Total	56,214	14,054	**70,268**

Table III.3: Data Distribution Between Training and Testing Sets

5.5 Classification

Table III.4 presents the performance metrics of various machine learning models. The Logistic Regression (LR) model achieved the highest accuracy of 82.4% and an F1 score of 81.3%, slightly outperforming the LinearSVC model, which recorded an accuracy of 82.0% and an F1 score of 80.9%. The Multinomial Naive Bayes (MNB) and Decision Tree (DT) models demonstrated moderate performance, with the MNB achieving an accuracy of 68.4% and an F1 score of 59.3, and the DT attaining an accuracy of 67.3% and an F1 score of 67.1%. Notably, the performance of these two classifiers declined further when using the TF-IDF representation.

Model	BoW				TF-IDF			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
MNB	0.684	0.678	0.587	0.593	0.588	0.743	0.439	0.402
LinearSVC	0.794	0.776	0.785	0.780	0.820	0.810	0.809	0.809
LR	0.816	0.801	0.812	0.805	0.824	0.817	0.810	0.813
DT	0.673	0.663	0.681	0.671	0.639	0.624	0.641	0.631

Table III.4: ML-Based Models

For deep learning models, and the transfer learning based models, hyperparameters were manually tuned to optimize performance (refer to the Table III.5).

Model	Optimal Configuration
CNN	`{'batch_size': 256, 'epochs': 20, 'learning_rate': 5e-4}`
LSTM	`{'batch_size': 128, 'epochs': 30, 'learning_rate': 8e-4}`
DistilBERT	`{'batch_size': 6, 'epochs': 3, 'learning_rate': 5e-5}`

Table III.5: Optimal Configuration

The LSTM and CNN, performed exceptionally well, especially when using the pre-trained GloVe embeddings, achieving 88% and 86.4% accuracy, respectively. However, it was observed that the performance of the CNN model decreased when using Google News Word2Vec embeddings compared to GloVe embeddings. The detailed results can be found in Table III.6.

Model	GloVe				Word2Vec			
	Accuracy	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
LSTM	0.880	0.886	0.861	0.872	0.879	0.889	0.859	0.871
CNN	0.864	0.866	0.847	0.855	0.824	0.812	0.812	0.812

Table III.6: DL-Based Models

However, the most efficient classifier is DistilBERT, which slightly outperforms the LSTM model, achieving an accuracy of 89% and an F1 score of 88.3% (Table III.7).

Model	Accuracy	Precision	Recall	F1 Score
DistilBERT	0.890	0.897	0.872	0.883

Table III.7: Fine-tuned DistilBERT

5.6 Deployment

Figure III.4 illustrates the general web interface, while Figure III.5 provides an example of testing the DistilBERT model.

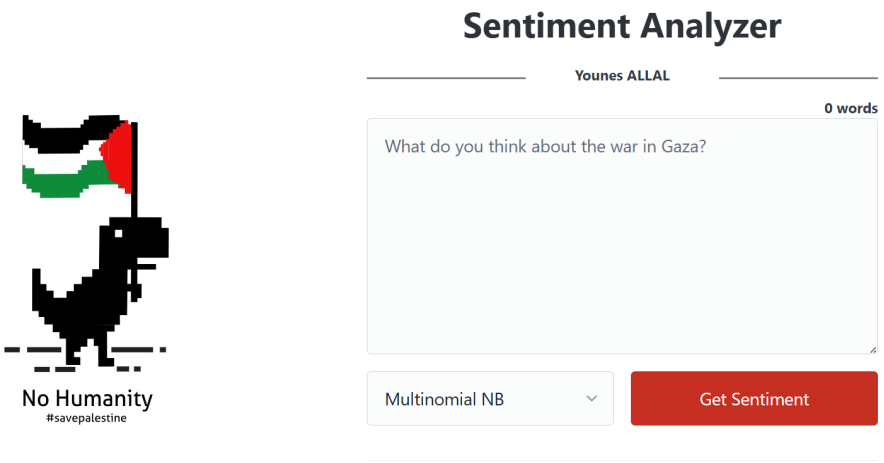

Figure III.4: Web Interface

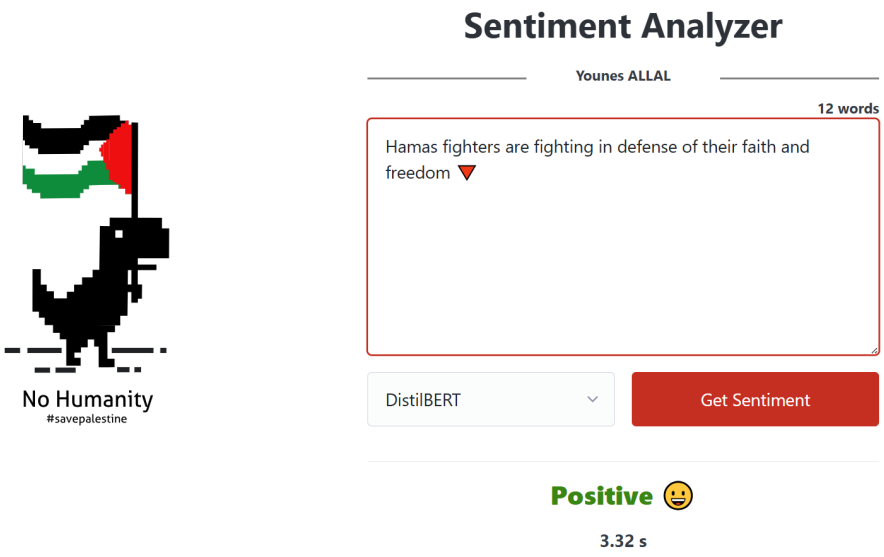

Figure III.5: Web Interface (Testing with DistilBERT)

6 Additional Experiments

After the evaluation of the models, three experiments were carried out in different phases:

6.1 Annotation: AFINN

The first experiment involved utilizing an alternative lexicon tool for the annotation process. Specifically, we employed the AFINN lexicon, which, as indicated in [43], demonstrates superior performance when handling negative comments. Code snippet III.16 represents the implementation of AFINN using the `afinn` Python library, while Figure III.6 presents a heatmap comparing the annotations provided by VADER with those generated by AFINN.

```
from afinn import Affin

score = Afinn(emoticons=True).score(text)

if score > 0:
    sentiment = 'Positive'
elif score < 0:
    sentiment = 'Negative'
else:
    sentiment = 'Neutral'
```

Code Snippet III.16: Text Annotation (AFINN)

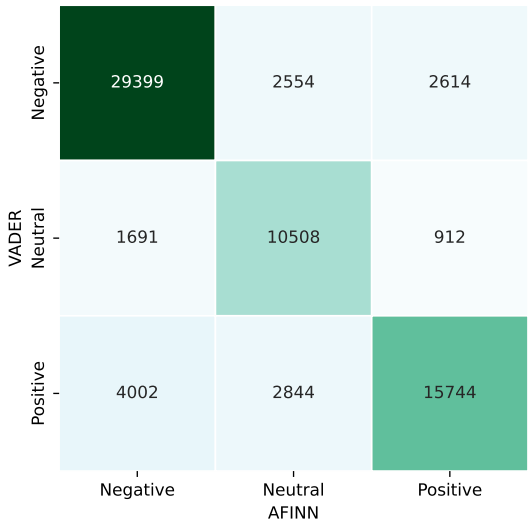

Figure III.6: VADER vs AFINN

After carrying out the other steps, the classification results showed an improvement in both accuracy and F1 score, as shown in Tables III.8, III.9, and III.10.

Model	BoW				TF-IDF			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
MNB	0.695	0.683	0.610	0.615	0.573	0.747	0.427	0.389
LinearSVC	0.853	0.832	0.828	0.830	0.864	0.849	0.837	0.843
LR	0.846	0.825	0.827	0.826	0.848	0.836	0.820	0.827
DT	0.706	0.685	0.694	0.689	0.674	0.650	0.659	0.654

Table III.8: ML-Based Models (AFINN)

Model	GloVe				Word2Vec			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
LSTM	0.898	0.889	0.876	0.881	0.900	0.887	0.882	0.884
CNN	0.882	0.870	0.858	0.863	0.848	0.828	0.827	0.827

Table III.9: DL-Based Models (AFINN)

Model	Accuracy	Precision	Recall	F1 Score
DistilBERT	0.911	0.900	0.895	0.897

Table III.10: Fine-tuned DistilBERT (AFINN)

However, despite this improvement, a closer examination of the dataset and the annotation methods of both AFINN and VADER revealed that the accuracy of these tools is not perfect. This observation highlights the crucial role of experts in the field of data science. While it is possible to collect large datasets, the lack of expert-led, objective annotation reduces the robustness of our studies. Therefore, the involvement of domain experts is essential to ensure the validity and accuracy of data annotations, highlighting a critical aspect for future research efforts.

6.2 Preprocessing: Removing Stop Words

Stop words represent the frequently used words in a specific language, such as ”an”, ”the”, and ”of” in English. Typically, researchers aim to remove these words during the preprocessing phase of a classification task, as they are often considered to lack significant semantic content.

In our study, we experimented with the removal of stop words and subsequently conducted our classification, with the function detailed in Code Snippet III.17.

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

def remove_stop_words(comment):
    stop_words = set(stopwords.words('english'))
    tokens = nltk.word_tokenize(comment)
    filtered_comment = [word.lower() for word in tokens if word.lower()
    ↪  not in stop_words]
    return ' '.join(filtered_comment)
```

Code Snippet III.17: Removing Stop Words

The results indicate that for traditional machine learning classifiers, the removal of stop words generally does not produce substantial differences in performance. An exception to this was observed with the Decision Tree (DT) classifier, where the accuracy and F1 score significantly improved from 63.9% and 63.1% to 71.4% and 71.7%, respectively, after stop words were removed.

Conversely, the performance of deep learning-based classifiers, and the fine-tuned DistilBERT model, declined upon the removal of stop words, as evidenced in Table III.11 and Table III.12.

Model	**GloVe**				**Word2Vec**			
	Accuracy	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
LSTM	0.850	0.858	0.834	0.845	0.848	0.857	0.831	0.843
CNN	0.839	0.842	0.824	0.832	0.801	0.797	0.785	0.789

Table III.11: DL-Based Models (After Removing Stop Words)

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
DistilBERT	0.853	0.861	0.837	0.848

Table III.12: Fine-tuned DistilBERT (After Removing Stop Words)

This result suggests that removing stop words, while useful in minimising feature space, is not always beneficial in terms of performance.

6.3 Classification: Hyperparameter Tuning for ML

When training our models, we manually tuned the hyperparameters of both the deep learning models (LSTM and CNN) and the DistilBERT model. However, we employed machine learning-based classifiers with their default hyperparameters, without any modifications.
In this experiment, our objective was to optimize the hyperparameters of these machine learning-based models (MNB, LinearSVC, LR, and DT). To achieve this, we utilized the `GridSearchCV` method from Scikit-learn, which systematically trains the models with various hyperparameter combinations to identify the optimal set. This technique leverages Cross Validation as a means of evaluating these combinations.
For our experiment, we employed `StratifiedKFold` (with $k = 3, 5$, and 10) for cross-validation instead of the default `KFold`. This approach ensures that the class distribution is maintained between the training and validation sets. We initiated the process by defining a list of hyperparameters for the model and began training. The best set of hyperparameters identified through this process was subsequently used to retrain the classifiers on the entire training set (refer to the Code Snippet III.18).

```
from sklearn.model_selection import GridSearchCV, StratifiedKFold
from sklearn.naive_bayes import MultinomialNB

hyparameters = {
    'alpha' : np.arange(0.1,2.1, 0.1),
    'fit_prior' : [True, False]
    }

for k in [3, 5, 10]:
    # Initialize StratifiedKFold with k splits
    cv = StratifiedKFold(n_splits=k, shuffle=True, random_state=1)
    # Initialize GridSearchCV
    clf = GridSearchCV(MultinomialNB(), param_grid=hyparameters, cv=cv,
     ↪  scoring='f1_macro', refit=True)
    # Fit the model using GridSearchCV
    clf.fit(features_train, y_train)
    # Print the best F1 score and hyperparameters
    print(clf.best_score_)
    print(clf.best_params_)
```

Code Snippet III.18: Hyperparameter Tuning (e.g, MNB)

Following this, we tested our machine learning models on the testing set. The results, as

detailed in Table III.13 and Table III.14, illustrate the best hyperparameters, and demonstrate a significant improvement in the performance of all models.

Model	Best hyperparameters (BoW, TF-IDF)
MNB	{'alpha': 0.4, 'fit_prior': False}, {'alpha': 0.2, 'fit_prior': False}
LinearSVC	{'C': 1, 'dual': True, 'loss': 'hinge', 'penalty': 'l2'}, {'C': 1, 'dual': True, 'loss': 'hinge', 'penalty': 'l2'}
LR	{'C': 1, 'dual': False, 'penalty': 'l1', 'solver': 'liblinear'}, {'C': 1, 'dual': False, 'penalty': 'l1', 'solver': 'saga'}
DT	{'criterion': 'gini', 'min_samples_split': 500, 'splitter': 'best'}, {'criterion': 'gini', 'min_samples_split': 500, 'splitter': 'best'}

Table III.13: Best Hyperparameters

Model	BoW		TF-IDF	
	Accuracy	F1 Score	Accuracy	F1 Score
MNB	0.709	0.680	0.702	0.665
LinearSVC	0.834	0.826	0.835	0.827
LR	0.834	0.825	0.840	0.832
DT	0.682	0.679	0.648	0.644

Table III.14: ML-Based Models (After Hyperparameter Tuning)

7 Conclusion

In this chapter, we presented the practical application of our methodology, providing detailed explanations and illustrative code snippets to elucidate each step of the process. We concluded with a comprehensive analysis of the results obtained, highlighting the effectiveness and efficiency of the implemented approaches.

GENERAL CONCLUSION

In the context of the alarming political and humanitarian situation in the Gaza Strip, which has persisted since last October, this thesis addresses one of the most popular natural language processing applications in recent years, sentiment analysis, which examines the global public opinion expressed on social media regarding the recent conflict.

Our objective was to compare different automated approaches to sentiment polarity classification to identify the most effective method and to examine whether data labeling impacts the classification process. Additionally, we aimed to determine if the language used on social media affects our workflow.

To achieve this, we collected public comments from the popular website Reddit, focusing on those related to the recent conflict. This process was followed by the standard steps of sentiment analysis, including labeling the data (positive, negative, and neutral), preprocessing to remove the noise typical of social media comments, and representing these comments in a form understandable by machine classifiers. Subsequently, we compared seven classifiers: four traditional machine learning models, two deep learning models, and we fine-tuned a transformer-based model. We evaluated their performance and conducted three different experiments, each focusing on a specific aspect for a more comprehensive comparison.

The results indicated that data labeling significantly impacts classifier effectiveness. Despite our success in gathering extensive data using web scraping techniques, the need for domain experts remains crucial.

Furthermore, the challenges of handling the informal language prevalent on social media became evident, especially when dealing with abbreviations, stop words, repeated characters, spelling errors, and so forth.

In terms of performance, transfer learning proved superior, with the fine-tuned DistilBERT achieving an accuracy of 89%, outperforming the deep learning classifiers LSTM and CNN, which achieved accuracies of 88% and 86.4%, respectively. These were followed by the four traditional classifiers, with LinearSVC leading this group with an accuracy of 84% after the

process of hyperparameter tuning.

Despite our comprehensive approach, there are many areas for improvement and further research, including:

- Employing multilingual sentiment analysis.
- Transitioning from sentiment polarity analysis to emotion recognition, a significant challenge in this field, particularly with textual data.
- Comparing the previous approaches using an expert-labeled dataset.
- Working with a balanced dataset across its different categories.
- Further exploring feature engineering and applying its techniques.
- Employing other methodologies such as hybrid approaches and lexicon-based methods, comparing them with different classifiers and techniques, and attempting to fine-tune GPT to do classification tasks instead of generation tasks.
- Enhancing the efficiency and response speed of the different classifiers.

This study highlights the importance of these aspects and it will set the stage for future research in this evolving field.

BIBLIOGRAPHY

[1] Abadi, Martín et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." In: *arXiv preprint arXiv:1603.04467* (2016).

[2] Adel, Hadeer et al. "Improving Crisis Events Detection Using DistilBERT with Hunger Games Search Algorithm." In: *Mathematics* 10.3 (2022). ISSN: 2227-7390. DOI: 10.3390/math10030447. URL: https://www.mdpi.com/2227-7390/10/3/447.

[3] Ahmad, Munir et al. "Hybrid Tools and Techniques for Sentiment Analysis: A Review." In: *International Journal of Multidisciplinary Sciences and Engineering* 8 (June 2017), pp. 31–38.

[4] Ahmed, Hafiz et al. "Sentiment Analysis of Online Food Reviews using Big Data Analytics." In: *İlköğretim Online* 20 (Apr. 2021), pp. 827–836. DOI: 10.17051/ilkonline.2021.02.93.

[5] Alamoudi, Eman Saeed and Alghamdi, Norah Saleh. "Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings." In: *Journal of Decision Systems* 30.2-3 (2021), pp. 259–281. DOI: 10.1080/12460125.2020.1864106. URL: https://doi.org/10.1080/12460125.2020.1864106.

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

[8] Borg, Anton and Boldt, Martin. "Using VADER sentiment and SVM for predicting customer response sentiment." In: *Expert Systems with Applications* 162 (2020), p. 113746. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113746. URL: https://www.sciencedirect.com/science/article/pii/S0957417420305704.

[9] Cambria, Erik. "Affective Computing and Sentiment Analysis." In: *IEEE Intelligent Systems* 31.2 (2016), pp. 102–107. DOI: 10.1109/MIS.2016.31.

[10] Chandrasekaran, Ganesh et al. "Visual Sentiment Analysis Using Deep Learning Models with Social Media Data." In: *Applied Sciences* 12.3 (2022). ISSN: 2076-3417. DOI: **10.3390/app12031030**. URL: https://www.mdpi.com/2076-3417/12/3/1030.

[11] Chauhan, Priyavrat, Sharma, Nonita, and Sikka, Geeta. "The emergence of social media data and sentiment analysis in election prediction." In: *Journal of Ambient Intelligence and Humanized Computing* 12.2 (Feb. 2021), pp. 2601–2627. ISSN: 1868-5145. DOI: **10.1007/s12652-020-02423-y**. URL: https://doi.org/10.1007/s12652-020-02423-y.

[12] Chintalapudi, Nalini, Battineni, Gopi, and Amenta, Francesco. "Sentimental Analysis of COVID-19 Tweets Using Deep Learning Models." In: *Infectious Disease Reports* 13.2 (2021), pp. 329–339. ISSN: 2036-7449. DOI: **10.3390/idr13020032**. URL: https://www.mdpi.com/2036-7449/13/2/32.

[14] Devlin, Jacob et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *arXiv preprint arXiv:1810.04805* (2018).

[16] Dzedzickis, Andrius, Kaklauskas, Artūras, and Bucinskas, Vytautas. "Human Emotion Recognition: Review of Sensors and Methods." In: *Sensors* 20.3 (2020). ISSN: 1424-8220. DOI: **10.3390/s20030592**. URL: https://www.mdpi.com/1424-8220/20/3/592.

[17] Gannouni, Sofien et al. "Emotion detection using electroencephalography signals and a zero-time windowing-based epoch estimation and relevant electrode identification." In: *Scientific Reports* 11.1 (Mar. 2021), p. 7071. ISSN: 2045-2322. DOI: **10.1038/s41598-021-86345-5**. URL: https://doi.org/10.1038/s41598-021-86345-5.

[18] Grundmann, Felix, Epstude, Kai, and Scheibe, Susanne. "Face masks reduce emotion-recognition accuracy and perceived closeness." In: *PLOS ONE* 16.4 (Apr. 2021), pp. 1–18. DOI: **10.1371/journal.pone.0249792**. URL: https://doi.org/10.1371/journal.pone.0249792.

[19] Hassan, Syed Zohaib et al. "Visual Sentiment Analysis from Disaster Images in Social Media." In: *Sensors* 22.10 (2022). ISSN: 1424-8220. DOI: **10.3390/s22103628**. URL: https://www.mdpi.com/1424-8220/22/10/3628.

[20] Hutto, C. and Gilbert, Eric. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text." In: *Proceedings of the International AAAI Conference on Web and Social Media* 8.1 (May 2014), pp. 216–225. DOI: **10.1609/icwsm.v8i1.14550**. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[22] Issa, Dias, Fatih Demirci, M., and Yazici, Adnan. "Speech emotion recognition with deep convolutional neural networks." In: *Biomedical Signal Processing and Control* 59 (2020), p. 101894. ISSN: 1746-8094. DOI: **https://doi.org/10.1016/j.bspc.2020.101894**. URL: https://www.sciencedirect.com/science/article/pii/S1746809420300501.

[23] Liu, Bing. *Sentiment analysis and opinion mining*. Springer Nature, 2022.

[24] Ma, Baojun, Yuan, Hua, and Wu, Ye. "Exploring performance of clustering methods on document sentiment analysis." In: *Journal of Information Science* 43.1 (2017), pp. 54–74. DOI: **10.1177/0165551515617374**. URL: https://doi.org/10.1177/0165551515617374.

[25] Marini, Marco et al. "The impact of facemasks on emotion recognition, trust attribution and re-identification." In: *Scientific Reports* 11.1 (Mar. 2021), p. 5577. ISSN: 2045-2322. DOI: **10.1038/s41598-021-84806-5**. URL: https://doi.org/10.1038/s41598-021-84806-5.

[26] Matalon, Yogev et al. "Using sentiment analysis to predict opinion inversion in Tweets of political communication." In: *Scientific Reports* 11.1 (Mar. 2021), p. 7250. ISSN: 2045-2322. DOI: **10.1038/s41598-021-86510-w**. URL: https://doi.org/10.1038/s41598-021-86510-w.

[28] Mikolov, Tomas et al. "Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781* (2013).

[29] Mishev, Kostadin et al. "Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers." In: *IEEE Access* 8 (2020), pp. 131662–131682. DOI: **10.1109/ACCESS.2020.3009626**.

[30] Muhammad, Putra, Kusumaningrum, Retno, and Wibowo, Adi. "Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews." In: *Procedia Computer Science* 179 (Jan. 2021), pp. 728–735. DOI: **10.1016/j.procs.2021.01.061**.

[31] Nandurkar, Tanmay et al. "Sentiment Analysis Towards Russia - Ukrainian Conflict: Analysis of Comments on Reddit." In: *2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP)*. 2023, pp. 1–6. DOI: **10.1109/ICETET-SIP58143.2023.10151571**.

[32] Paszke, Adam et al. "Pytorch: An imperative style, high-performance deep learning library." In: *Advances in neural information processing systems* 32 (2019).

[33] Pedregosa, F. et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[34] Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. "GloVe: Global Vectors for Word Representation." In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[35] Pipalia, Keval, Bhadja, Rahul, and Shukla, Madhu. "Comparative Analysis of Different Transformer Based Architectures Used in Sentiment Analysis." In: *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*. 2020, pp. 411–415. DOI: **10.1109/SMART50582.2020.9337081**.

[36] Prasad, Dilip K. et al. "Sentiment analysis using EEG activities for suicidology." In: *Expert Systems with Applications* 103 (2018), pp. 206–217. ISSN: 0957-4174. DOI: **https://doi.org/10.1016/j.eswa.2018.03.011**. URL: https://www.sciencedirect.com/science/article/pii/S0957417418301507.

[37] Qaisar, Saeed Mian. "Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory." In: *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*. 2020, pp. 1–4. DOI: **10.1109/ICCIS49240.2020.9257657**.

[38] Reshi, Aijaz Ahmad et al. "COVID-19 Vaccination-Related Sentiments Analysis: A Case Study Using Worldwide Twitter Dataset." In: *Healthcare* 10.3 (2022). ISSN: 2227-9032. DOI: **10.3390/healthcare10030411**. URL: https://www.mdpi.com/2227-9032/10/3/411.

[39] Rubenstein, Herbert and Goodenough, John B. "Contextual correlates of synonymy." In: *Commun. ACM* 8.10 (Oct. 1965), pp. 627–633. ISSN: 0001-0782. DOI: **10.1145/365628.365657**. URL: https://doi.org/10.1145/365628.365657.

[40] Sanh, Victor et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." In: *arXiv preprint arXiv:1910.01108* (2019).

[42] Sepúlveda, Axel et al. "Emotion Recognition from ECG Signals Using Wavelet Scattering and Machine Learning." In: *Applied Sciences* 11.11 (2021). ISSN: 2076-3417. DOI: **10.3390/app11114945**. URL: https://www.mdpi.com/2076-3417/11/11/4945.

[43] Al-Shabi, Mohammed. "Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining." In: (Aug. 2020).

[44] Shen, Fangyao et al. "Multi-Scale Frequency Bands Ensemble Learning for EEG-Based Emotion Recognition." In: *Sensors* 21 (Feb. 2021), p. 1262. DOI: **10.3390/s21041262**.

[46] Sohangir, Sahar et al. "Big Data: Deep Learning for financial sentiment analysis." In: *Journal of Big Data* 5 (Jan. 2018). DOI: **10.1186/s40537-017-0111-6**.

[47] Soundariya, R.S. and Renuga, R. "Eye movement based emotion recognition using electrooculography." In: *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. 2017, pp. 1–5. DOI: **10.1109/IPACT.2017.8245212**.

[50] The pandas development team. *pandas-dev/pandas: Pandas*. Version v2.2.2. Apr. 2024. DOI: **10.5281/zenodo.10957263**. URL: https://doi.org/10.5281/zenodo.10957263.

[51] Vajjala, S. et al. *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*. O'Reilly Media, 2020. ISBN: 9781492054054.

[52] Villavicencio, Charlyn et al. "Twitter Sentiment Analysis towards COVID-19 Vaccines in the Philippines Using Naïve Bayes." In: *Information* 12.5 (2021). ISSN: 2078-2489. DOI: **10.3390/info12050204**. URL: https://www.mdpi.com/2078-2489/12/5/204.

[53] Wadhwani, Ganesh Kumar et al. "Sentiment Analysis and Comprehensive Evaluation of Supervised Machine Learning Models Using Twitter Data on Russia–Ukraine War." In: *SN Computer Science* 4.4 (Apr. 2023), p. 346. ISSN: 2661-8907. DOI: **10.1007/s42979-023-01790-5**. URL: https://doi.org/10.1007/s42979-023-01790-5.

[54] Waheeb, Samer Abdulateef, Khan, Naseer Ahmed, and Shang, Xuequn. "Topic Modeling and Sentiment Analysis of Online Education in the COVID-19 Era Using Social Networks Based Datasets." In: *Electronics* 11.5 (2022). ISSN: 2079-9292. DOI: **10.3390/electronics11050715**. URL: https://www.mdpi.com/2079-9292/11/5/715.

[55] Wankhade, Mayur, Rao, Annavarapu Chandra Sekhara, and Kulkarni, Chaitanya. "A survey on sentiment analysis methods, applications, and challenges." In: *Artificial Intelligence Review* 55.7 (Oct. 2022), pp. 5731–5780. ISSN: 1573-7462. DOI: **10.1007/s10462-022-10144-1**. URL: https://doi.org/10.1007/s10462-022-10144-1.

[56] Yasen, Mais and Tedmori, Sara. "Movies Reviews Sentiment Analysis and Classification." In: *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. 2019, pp. 860–865. DOI: **10.1109/JEEIT.2019.8717422**.

WEBOGRAPHY

[6] *Announcment of shutting down the free API of twitter*. https://x.com/XDevelopers/status/1621026986784337922. (Accessed on 21/05/2024).

[13] *Contraction | English meaning - Cambridge Dictionary*. https://dictionary.cambridge.org/dictionary/english/contraction. (Accessed on 23/05/2024).

[15] *DistilBERT-BASE-UNCASED | Hugging Face*. https://huggingface.co/distilbert/distilbert-base-uncased. (Accessed on 15/06/2024).

[21] *Israel has lost the war of public opinion | Opinions | Al Jazeera*. https://www.aljazeera.com/opinions/2023/11/30/israel-has-lost-the-war-of-public-opinion. (Accessed on 22/06/2024).

[27] *McDonald's, Starbucks See New Losses From Middle East Boycotts - Business Insider*. https://www.businessinsider.com/mcdonalds-starbucks-see-new-losses-from-middle-east-boycotts-2024-5. (Accessed on 22/06/2024).

[41] *Semi-Supervised Learning, Explained*. https://www.altexsoft.com/blog/semi-supervised-learning/. (Accessed on 21/06/2024).

[45] *Slang | English meaning - Cambridge Dictionary*. https://dictionary.cambridge.org/dictionary/english/slang. (Accessed on 23/05/2024).

[48] *Starbucks partner 'to cut thousands of jobs' over Gaza-linked boycotts | Business News | Sky News*. https://news.sky.com/story/starbucks-partner-to-cut-thousands-of-jobs-over-gaza-linked-boycotts-13087738. (Accessed on 22/06/2024).

[49] *Text Classification | Hugging Face*. https://huggingface.co/docs/transformers/tasks/sequence_classification. (Accessed on 15/06/2024).

APPENDIX A

DATASET

The dataset can be accessed through the provided GitHub account link or via the accompanying QR code.
The dataset is divided into two parts: training and test datasets, both available in CSV format. Each dataset comprises two columns: `id_comment` and `label`.
For extracting comments, the Python library PRAW (Python Reddit API Wrapper) can be utilized. Detailed instructions are available on the same GitHub page.

Figure A.1: Dataset (https://github.com/unus-all/sentiment-analysis)

APPENDIX B

HYPERPARAMETER TUNING

Here are the detailed results obtained during hyperparameter tuning (refer to the third chapter (subsection 6.3)).

Model	Features	GridSearch Cross validation (StratifiedKFold)		
		K	Best Parameters	GS - F1
MNB	TF-IDF	3	{'alpha': 0.1, 'fit_prior': False}	0.644
		5		0.649
		10	{'alpha': 0.2, 'fit_prior': False}	0.656
	BoW	3	{'alpha': 0.3, 'fit_prior': False}	0.658
		5	{'alpha': 0.4, 'fit_prior': False}	0.665
		10		0.669
LinearSVC	TF-IDF	3	{'C': 1, 'dual': False, 'loss': 'squared_hinge', 'penalty': 'l1'}	0.8126
		5	{'C': 1, 'dual': True, 'loss': 'hinge', 'penalty': 'l2'}	0.8173
		10		0.8225
	BoW	3	{'C': 0.1, 'dual': False, 'loss': 'squared_hinge', 'penalty': 'l1'}	0.8121
		5		0.8177
		10	{'C': 1, 'dual': True, 'loss': 'hinge', 'penalty': 'l2'}	0.8217
LR	TF-IDF	3	{'C': 1, 'dual': False, 'penalty': 'l1', 'solver': 'saga'}	0.8204
		5		0.8264
		10		0.8288
	BoW	3	{'C': 1, 'dual': False, 'penalty': 'l1', 'solver': 'liblinear'}	0.816
		5		0.8217
		10		0.8243
DT	TF-IDF	3	{'criterion': 'gini', 'min_samples_split': 500, 'splitter': 'best'}	0.6231
		5		0.6314
		10		0.6368
	BoW	3		0.6709
		5		0.6749
		10		0.6815

Figure B.1: Results of GridSearchCV