People's Democratic Republic Of Algeria

Ministry of Higher Education and Scientific Research

University of Echahid Hamma Lakhder-El ouad

Faculty of Exact Sciences

Computer Science department

End of study memory

Presented for the Diploma of

ACADEMIC MASTER

Domain: **Mathematics and Computer Science**

Industry: **Computer Science**

Specialty: **Distributed Systems and Artificial Intelligence**

Theme

Opinion Mining From Text Based On Machine Learning Approach

Presented by:

Bechoua Abla

Halouadji Abir

DR. Khaoula belila Supervisor Univ. El Oued

DR. Boucherit Ammar president Univ. El Oued

DR. Soltani Khaled Examiner Univ. El Oued

Academic year

2023/2024

ABSTRACT

In this work we describe our attempt to evaluate the performance of machine learning methods in analyzing sentiment and opinions expressed in text data with SemEval-2017 Task 4 “Sentiment Analysis in Twitter”. We chose subtask A for Message-Level Sentiment Classification, We use supervised maching learning algrothims with word embeddings techniques such as Tf-Idf , Bow and Word2Vec. Also, we present a text processing method suitable for social network messages, which performs tokenization, word lemmatization and more. This work can be extremely valuable for businesses and organizations that want to understand customer feedback, track brand reputation, or analyze public perception of certain topic

Keywords: Opinion Mining, Sentiment Analysis, Emotions, Natural Language Processing.

RÉSUMÉ

Dans ce travail, nous décrivons notre tentative d'évaluer les performances des méthodes d'apprentissage automatique dans l'analyse des sentiments et des opinions exprimés dans les données textuelles avec SemEval-2017 Task 4"Sentiment Analysis in Twitter" . Nous avons choisi la "Subtask A" pour la classification des sentiments au niveau du message . Nous utilisons des algorithmes d'apprentissage par machine supervisé avec des techniques d'intégration de mots telles que Tf-Idf, Bow et Word2Vec.Nous présentons également une méthode de traitement de texte adaptée aux messages des réseaux sociaux, qui effectue la tokenisation, la lemmatisation des mots et bien plus encore. ce travail peut être extrêmement utile pour les entreprises et les organisations qui souhaitent comprendre les commentaires des clients, suivre la réputation de la marque ou analyser la perception du public sur certains sujets .

Mots-clés : Exploration d'opinions, analyse des sentiments, émotions, traitement du langage naturel.

الملخص

في هذا العمل، نصف محاولتنا لتقييم أداء أساليب التعلم الآلي في تحليل المشاعر والآراء المعبر عنها في البيانات النصية بإستخدام $SemEval-2017Task4$ تحليل المشاعر في تويتر. لقد اخترنا $subtaskA$ لتصنيف المشاعر على مستوى الرسالة، ولقد إستخدمنا خوارزميات تعلم الآلة الخاضعة للرقابة مع تقنيات تضمين الكلمات مثل $Tf-Idf$ و Bow و $Word2Vec$.نقدم أيضًا طريقة معالجة نصية مناسبة لرسائل الشبكات الاجتماعية، والتي تقوم بالترميز وتحويل الكلمات و أكثر من ذلك . يمكن أن يكون هذا العمل مفيد للغاية للشركات والمؤسسات التي ترغب في فهم تعليقات العملاء أو تتبع سمعة العلامة التجارية أو تحليل التصور العام لموضوعات معينة.

الكلمات المفتاحية : استخراج اللآراء، تحليل المشاعر، العواطف، معالجة اللغات الطبيعية

KNOWLEDGMENT

I want to dedicate this to myself first, yes , to me , because I never gave up. I kep getting up after every failure, and I always believed in the saying ” Because your dreams do not sleep, you must awaken your will as well ”. Now I finally see the result of my faith. So I want to remind myself:continue your creativity, your excellence and always stay strong.

Secondly, to my dear parents, thank you for believing in me and my choices. Thank you for supporting me at every step. Praise and thanks be to Allah that you are present in my life, and I ask Allah to always protect you for me.

Finally, to everyone who stood by me and helped me present my project well, Especially to my supervisor **khaoula belila**.

To my friends, colleagues , and teachers ...

Abla

I’d like to express my sincere gratitude to our supervisor **khaoula belila** , for her invaluable guidance, support, and encouragement throughout this research. her expertise and insights was instrumental in shaping the direction and outcomes of this study. A heartfelt thank you goes to my family members and friends for their unwavering support during the challenging times of this study journey. In particular, I extend my deepest gratitude to my parents, **Mosbah** and **Nacira**. Their patience and encouragement have been a constant source of motivation.

Abir

Contents

List of Abbreviations

ANNs Artificial Neural Networks.

BERT Bidirectional Encoder Representation from Transformers.

BoW Bag of Words..

DL Deep Learning.

LR Logistic Regression.

ME Maximum Entropy.

ML Machine Learning.

MLM Masked Language Model.

NB Naive Bayes.

NLP Natural Language Processing..

SA Sentiment analysis.

SC Sentiment classification.

SVM Support Vector Machines.

TF-IDF Term Frequency-Inverse Document Frequency..

List of Figures

List of Tables

CHAPTER I

Foundations and Context

I.1 Introduction

In the realm of natural language processing, opinion mining, also known as sentiment analysis, acts like a detective. It sifts through written text, such as online reviews, to uncover the hidden opinions and emotions. This allows businesses to gain valuable customer insights by understanding how people feel about their products and services. It's like having a direct line to the customer's mind. To achieve this feat, opinion mining draws on various disciplines like machine learning, NLP, sociology, and psychology. Social media platforms like Twitter and Facebook have become treasure troves of user opinions, and businesses have been leveraging this data for close to a decade to refine their strategies and improve customer satisfaction.

I.2 Problem Statement

Machine learning models have proven valuable for various opinion mining tasks. This work investigates the effectiveness of state-of-the-art ML models, particularly those utilizing word embeddings, in addressing these tasks. We will implement these models on the SemEval-2017 Task 4 dataset to evaluate their performance

I.3 Aim and Objectives

I.3.1 Aim

To enhance the accuracy and robustness of sentiment analysis methods using machine learning approaches for extracting opinions from text data, focusing on developing more accurate and robust methods for machines to analyze sentiment and opinions expressed in text data.

I.3.2 Objectives

- Apply different word embedding methods with machine learning techniques.
- Evaluate and compare the effectiveness of different machine learning algorithms for sentiment analysis, aiming to identify the most suitable approach for the given task.

I.4 Outline

This section outlines the roadmap for our thesis

- Chapter 1: This chapter introduces our thesis topic, motivation, research goals, and a brief overview of the problem, while also diving into the technical background by exploring core principles and relevant concepts that underpin our work.
- Chapter 2: This chapter details the specific algorithms we'll be employing. Here, we'll explain the selection process and provide a summary of each chosen algorithm.
- Chapter 3:This chapter takes center stage, detailing the experiments we designed. we'll meticulously outline the steps involved in each experiment.
- Chapter 4: This chapter presents the results obtained from our experiments in a clear and organized manner, while the concluding chapter summarizes key takeaways, reiterates the importance of our research, and proposes future research directions.

I.5 Maching Learning

The field of machine learning has witnessed a surge in popularity in recent years, driven by the ever-increasing need for intelligent machines capable of solving complex problems. This rise

in interest can be attributed to our fascination with mimicking human capabilities, specifically how we learn. Humans acquire knowledge primarily through experience, an ability that machine learning [1]aims to replicate in machines. As research progresses, machine learning holds the promise of significantly reducing human effort. It allows machines to learn independently through past experiences, utilizing three main approaches: supervised learning, unsupervised learning, and reinforcement learning.

- **Supervised learning** : Supervised learning is a type of machine learning where a model is trained on labeled data, meaning each input comes with a corresponding output. The model learns to predict the output from the input. Common applications include classification (e.g., spam detection) and regression (e.g., predicting prices).
- **Unsupervised learning** :Unsupervised learning is a type of machine learning where the algorithm analyzes and finds patterns in data without labeled outputs. The main goal is to uncover hidden structures and relationships within the data. Common techniques include clustering (grouping similar data points) and dimensionality reduction (simplifying data by reducing its features). Applications of unsupervised learning include market segmentation, anomaly detection, and recommendation systems.

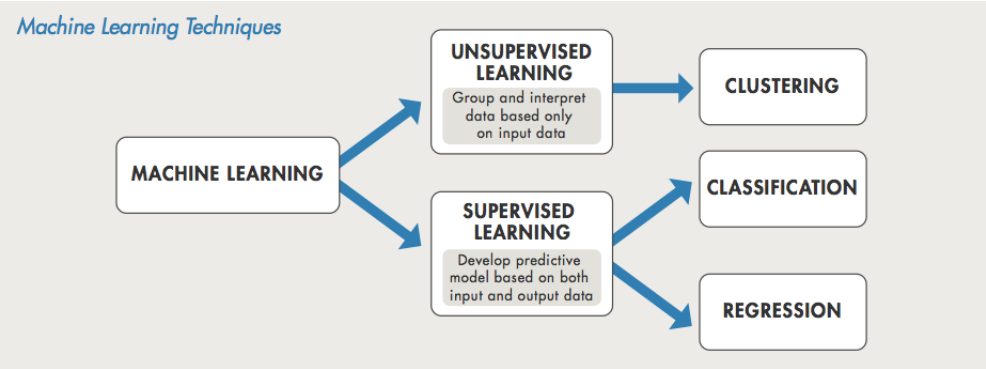

Figure I.1: Comparing supervised and unsupervised learning

- **Reinforcement learning** : Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards. It's about learning through trial and error to achieve long-term goals.

I.6 Traditional Sentiment classification techniques

Traditionally, sentiment analysis was handled using machine learning models like Naive Bayes (NB), Logistic Regression (LR), and Support Vector Machines (SVM) ,Maximum Entropy (ME). These models often utilize the 'bag of words' approach, treating each word as an individual unit and ignoring the context and sequence in which they occur.

- **Naive Bayes(NB):**

  This is a probabilistic classification technique [2]. This classifier performs well when applied to large datasets [3]. NB classifier computes posterior probability by using the formula

$$\text{posterior probability} = \frac{likelihood \times priorprobability}{evidence} \tag{I.1}$$

  Equivalently,

$$P(Class_i|_z) = \frac{P(_z|Class_i) \times P(Class_i)}{P(_z)} \tag{I.2}$$

  Where z represents the feature vector and $Class_i$ represents the i^{th} class. NB classifier

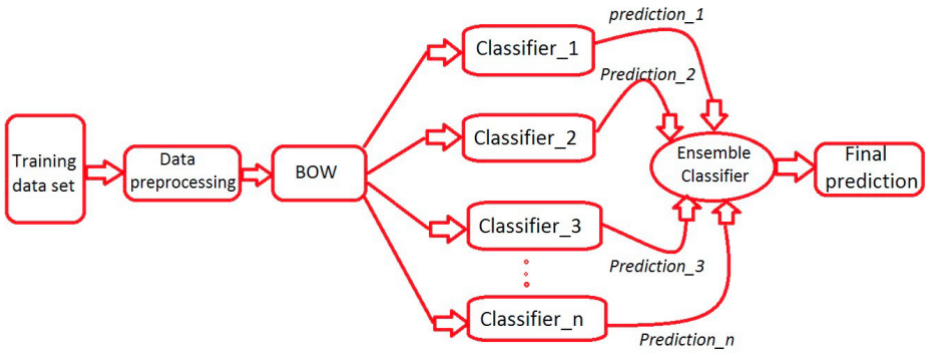

Figure I.2: . An overview of tweet sentiment classification approach using ensemble classifier

  makes an assumption that features are conditionally independent. Smoothing techniques

are used to eliminate undesirable effects.

- **Random Forest(RF):**

  Random Forest [4] is an ensemble learning algorithm that constructs multiple decision trees during training and combines their predictions for classification or regression tasks. The default values for parameters in scikit learns implementation of Random Forest are as follows:

  n_estimators: 100, **max_depth:** None , **min_samples_split:** 2 ,**min_samples_leaf:** 1 , **max_features:** ”auto”, **bootstrap:** True , **random_state:** None .

- **Support Vector Machine(SVM):**

  SVM is a supervised learning algorithm. It is also called a probabilistic classifier used for classification and regression tasks. It finds the optimal hyperplane to separate data points into classes or predict continuous outcomes, maximizing the margin between classes for robust performance. [5]

  the default values of parameters for SVM classifiers are tuned as:

  C: 1.0 , **kernel:** ”rbf” (radial basis function), **gamma:** ”scale” (depends on 1 / (n_features * X.var())), **degree:** 3 (degree of the polynomial kernel function), **coef0:** 0.0, **class**$_{w}$*eight* : $None(allclassesaretreatedequally)$, **probability:**$False$, **shrinking:** $True$.

- **Logistic Regression(LR):**

  Logistic Regression is a statistical method used for binary classification. It models the probability of an input belonging to a particular class using a logistic function. It’s simple, interpretable, and widely used in various fields. [6]. The parameter values of the LR classifiers are tuned as: C : .01 , max_iter :100.

I.7 Deep Learning

Deep learning is a subset of machine learning that involves training artificial neural networks with multiple layers. [1] It automatically learns hierarchical representations from data, revolutionizing fields like computer vision and natural language processing.

I.7.1 Deep Neural Network

Deep learning, a new frontier in machine learning, emulates the intricate architecture and functionality of the human brain. This innovative algorithm automatically discerns crucial features from data, facilitating the creation of complex models. Represented as a mathematical function f: X $\rightarrow$ Y, deep learning extends the capabilities of artificial neural networks (ANNs) by incorporating multiple hidden layers. These layers, as illustrated in Figure II.3:

- the input layer : where neurons receive input from variable X
- hidden layers : each training distinctive features from preceding layers, with deeper layers capturing more abstract representations
- the output layer : where neurons generate the final output based on processed information from the hidden layers.

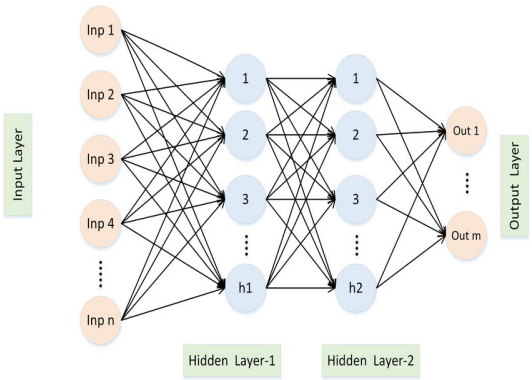

Figure I.3: Deep Neural Network

I.7.2 Convolutional Neural Network CNN

Convolutional Neural Network (CNN) is a specialized type of artificial neural network designed for processing and analyzing visual data, like images.[7] It's composed of layers that automatically learn features directly from pixel data, making it highly effective for tasks such as image classification and object detection.

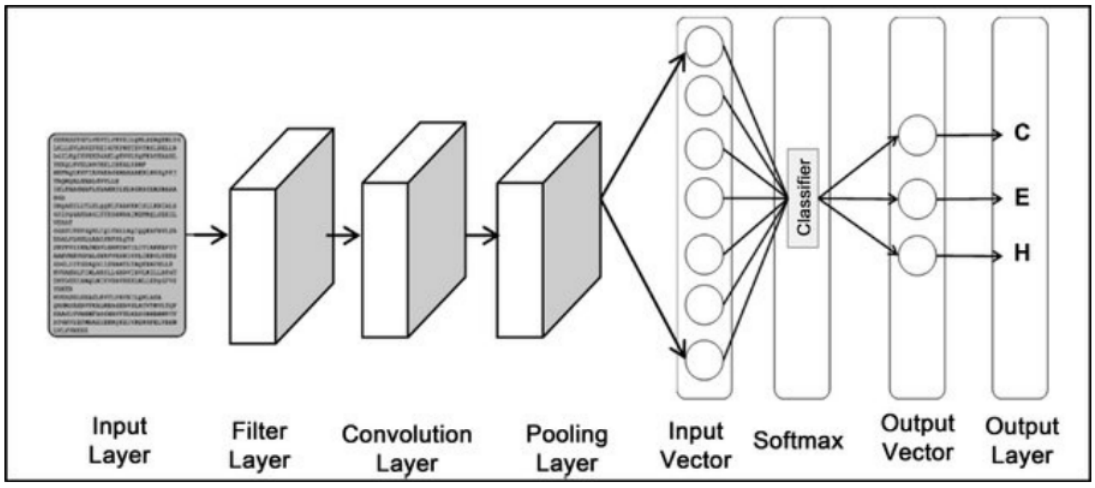

Figure I.4: Example of CNN architecture Natural language processing (NLP)

I.7.3 Recurrent neural network RNN

RNN are specialized neural networks designed for processing sequential data. They maintain an internal memory to capture information from past inputs, making them suitable for tasks like time series prediction,language modeling, and sentiment analysis. RNN utilize recurrent connections and share parameters across time steps. They are trained using backpropagation through time (BPTT) and have been applied successfully in various fields due to their ability to model sequential dependencies.

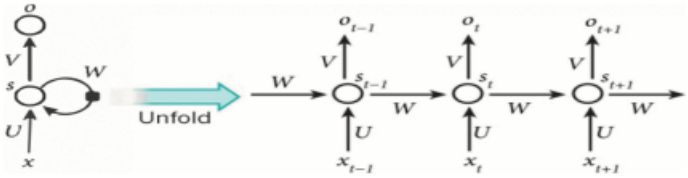

Figure I.5: RNN framework

the basic formula for an RNN described in another format:[8]:

$$a_t = f(h_{t-1}, x_t) \tag{I.3}$$

Where:

- $a_t$represents the output from the previous node.
- f is the activation function, which is typically the hyperbolic tangent (tanh) function.
- h_{t-1} is the hidden state from the previous time step.
- x_t denotes the input sequences $(x_0, x_{0*1}, x_2, \ldots, x_t)$.

RNNs capture the sequential nature of data, making them suitable for tasks like text analysis. However, they can be time-consuming due to their sequential processing nature. This drawback has been addressed to some extent by recurrent neural networks[9], which improve the time complexity by processing words in parallel. [10]

I.8 Machine Learning vs Deep Learning

Aspect	**Machine Learning**	**Deep Learning**
Data Requirements	Can work with smaller datasets	Large datasets required for optimal performance
Feature Engineering	Often requires significant feature engineering	Requires minimal feature engineering
Computational Resources	Less computationally intensive, often run on standard CPUs	High computational resources (e.g., GPUs)
Use Cases	Various tasks including classification, regression, and clustering	Image recognition, NLP, complex tasks

Table I.1: Comparison between Deep Learning and Machine Learning

I.9 Sentiment Analysis

Sentiment analysis (SA), a subfield of natural language processing (NLP) [11], has emerged as a prominent and rapidly growing area of research. It involves computationally analyzing opinions, sentiments, and subjectivity expressed within textual data. SA aims to identify the writer's emotional stance, whether positive or negative, by examining large volumes of text documents like comments, questions, and requests.

The applications of SA have blossomed across diverse fields. In the political arena, for example, it can be used to analyze sentiment in online forums, potentially aiding in predictions of election outcomes [12].Businesses can leverage SA to understand customer sentiment on social media,

potentially informing stock market forecasts [13]. Likewise, marketers can utilize SA to estimate sales of specific products based on online reviews [14].

However, a fundamental challenge in SA lies in accurately identifying the type and nature of sentences within a document. Documents often contain a mix of objective facts and subjective opinions. News articles, for example, typically focus on objective information, while product reviews are likely to express subjective opinions. Therefore, SA systems typically involve extracting and classifying sentences as subjective or objective before further analysis.

Subjectivity classification, a critical task in SA, involves labeling sentences based on whether they express an opinion or simply convey factual information. Sentences identified as objective are typically discarded as they offer little value for sentiment analysis. Following subjectivity classification, the remaining subjective sentences undergo sentiment classification. This task involves categorizing the sentiment polarity of the sentences as positive, negative, or neutral. A crucial task in SA is aspect or object-based extraction, which focuses on identifying the specific entities or topics being discussed in the sentiment. In some cases, it's also important to identify the opinion holder, which is the person or entity expressing the opinion.

There are three main approaches to sentiment classification: machine learning, lexicon-based, and hybrid approaches.

- **The Machine Learning Approach (ML) :** applies the famous ML algorithms and uses linguistic features as mentioned before

- **The lexicon-based approach [15]:** Lexicon-based sentiment analysis hinges on identifying a collection of opinion-related words used to assess the text's sentiment. This approach employs two main methods: *The dictionary-based approach* This method starts by identifying seed words that express opinions. It then searches a dictionary to find synonyms and antonyms of these seed words, expanding the lexicon of opinion-related terms. *The corpusbased approach* This method begins with a seed list of opinion words. It then leverages a large collection of text data (corpus) to discover additional opinion words with context-specific orientations. Statistical or semantic methods can be used for this purpose.

- **The hybrid Approach:** a common method in sentiment analysis, merges both the strengths of machine learning and lexicon-based approaches. Sentiment lexicons play a central role in most hybrid methods. These lexicons provide a foundation of sentiment-laden words,

while machine learning algorithms can then learn more complex patterns and relationships within the text data. This combined approach can potentially lead to more accurate and nuanced sentiment analysis compared to relying solely on one method. The various approaches and the most popular algorithms of SC are illustrated in Fig. I.6

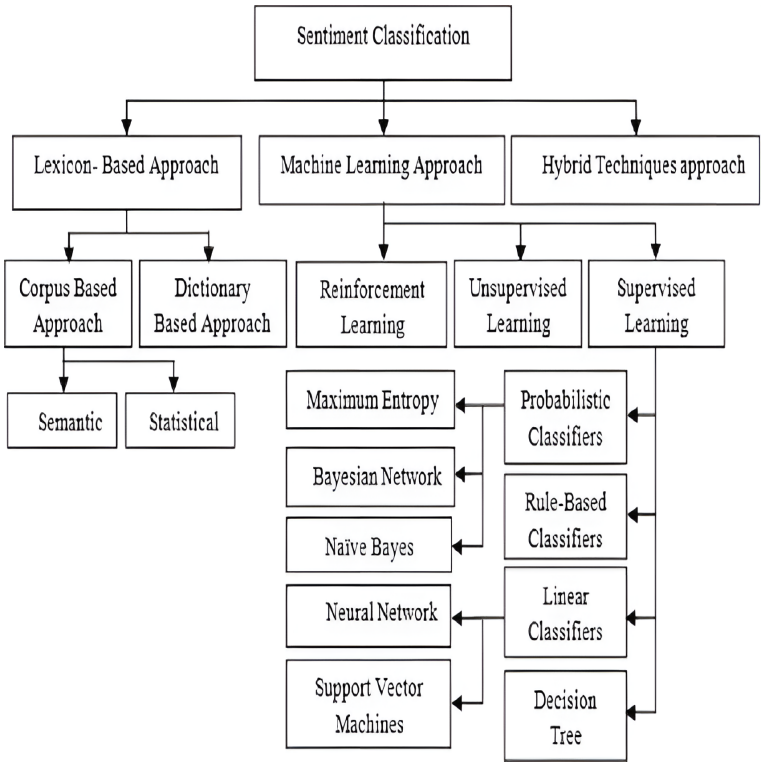

Figure I.6: Sentiment classification techniques.

I.10 Performance Measures

Evaluation metrics adopted within DL tasks play a crucial role in achieving the optimized classifer [16]. They are utilized within a usual data classifcation procedure through two main stages: training and testing. It is utilized to optimize the classifcation algorithm during the training stage. This means that the evaluation metric is utilized to discriminate and select the optimized solution, e.g., as a discriminator, which can generate an extra-accurate forecast of upcoming evaluations related to a specifc classifer. For the time being, the evaluation metric is utilized to measure the efciency of the created classifer, e.g. as an evaluator, within the model testing stage using hidden data. As given in Eq. II-6, *TN* and *TP* are defned as the number of negative and positive instances, respectively, which are successfully classifed. In addition, *FN* and *FP* are defned as the number of misclassifed positive and negative instances respectively. Next, some of the most well-known evaluation metrics [17] are listed below.

1. **Accuracy:** Calculates the ratio of correct predicted classes to the total number of samples evaluated .

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{I.4}$$

2. **Precision:** Utilized to calculate the positive patterns that are correctly predicted by all predicted patterns in a positive class.

$$Precision = \frac{TP}{TP + FP} \tag{I.5}$$

3. **Sensitivity or Recall:** Utilized to calculate the fraction of positive patterns that are correctly classifed.

$$Sensitivity = \frac{TP}{TP + FN} \tag{I.6}$$

4. **F1-Score:** Calculates the harmonic average between recall and precision rates.

$$F1_{scor} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{I.7}$$

I.11 Conclusion

This chapter has introduced our thesis topic, motivated its exploration, and outlined our research objectives, while also delving into the technical background essential for understanding our study. By establishing the purpose of our research and providing an overview of the problem, we have set the stage for a detailed examination of our methodologies and findings in subsequent chapters. Additionally, by exploring core principles and relevant concepts, we have built a strong theoretical foundation that supports our analysis. This comprehensive overview equips the reader with the necessary knowledge to grasp the complexities of our work as we embark on this academic journey to contribute meaningfully to our field of study.

CHAPTER II

Related Work

II.1 Introduction

In this chapter, we presented some of the selected studies for conducting sentiment analysis or opinion mining. Recently, different machine learning models (such as Logistic Regression , SVM , and Naive Bayes) have been used to improve the effi-ciency of sentiment analysis .

II.2 Literature Review and Related Studies

Since the late 20th century. several scholars have studied this tendency. like Bo Pang and Lillian Lee: Their work focused on applying machine learning techniques to analyze product reviews and movie reviews [18], respectively. One of their most important works is compare the effi-ciency of an automatic classifier based on dictionary with the classification by human jurors in a set of comments made by customers in Portuguese language[19]. The data consist of opinions of service users of one of the largest Brazilian online employment agencies. The performance evaluation of the classification models was done using kappa index and a confusion matrix. And Much of the research in unsupervised sentiment classification[20] using symbolic techniques makes use of ... [21][22] Turney used bag-of-words approach for sentiment analysis.
These researchers, along with others, laid the groundwork for the field's development.

Sentiment analysis has come a long way from its early days of simple counting postive and negative words. Where it appeared Minqing Hu and Bing Liu ,This duo is known for their re-

search on developing sentiment analysis techniques specifically for social media text[23]. They explored challenges like informal language, slang, and emojis, which can be tricky for sentiment analysis models trained on more formal text.

One approach uses sentiment lexicons [24]. These list words and their emotional associations (positive, negative, etc.). This sentiment information can then be combined with other techniques, like neural networks, to understand the overall feeling of a text.

research focused [25] on sentiment analysis in the financial domain .He addressed the specific challenges of analyzing financial news and messages, which often use technical language and can have subtle sentiment depending on context. He introduced the first graph neural network for Financial Sentiment Analysis using text and relationship features based on temporal, word, and entity information.

Other research involved Twitter sentiment analysis using Lexicon Method[26], Machine Learning Method and Their Combination.To quantify the performance of the main sentiment analysis methods over Twitter we run these algorithms on a benchmark Twitter dataset from the SemEval-2013 competition,task 2-B. The results[27] show that machine learning method based on SVM and Naive Bayes classifiers outperforms the lexion .

As suggested by some researchers, A Multi-view Ensemble [28] for Twitter Sentiment Analysis.Their system is a voting ensemble, where each base classifier is trained in a different feature space. The first space is a bag-of-words model and has a Linear SVM as base classifier. The second and third one spaces are two different strategies of combining word embeddings to represent sentences and use a Linear SVM and a Logistic Regressor as base classifiers.

Some studies have also appeared [29] that classify the book review on Amazon.com into positive reviews or negative reviews using a combination of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) algorithms.

This research [30] about analyzing data found on the web for Internet users , represented by quotes or comments by using various supervised or data-driven techniques to Sentiment analysis like Naïve Byes [31] , Maximum Entropy and SVM. As some studies [32] aims to perform sentiment analysis of real-time 2019 election twitter data using the feature selection model word2vec and the machine learning algorithm random forest for sentiment classification and improves the accuracy of sentiment analysis significantly compared to traditional methods such as BOW and TF-IDF.

In this research [33], they investigate a range of machine-learning strategies for solving sentiment

analysis challenges. Two datasets were analyzed with the models based on the term frequency-inverse document frequency (TF-IDF). A comparison study was conducted between each of the models to determine how they performed in experiments. Regarding accuracy and F1 score, logistic regression performs better than other algorithms.

Table II.1 shows the validation results on the historical test set for Subtask A [34]:

System	**2013**	**2014**	**2015**	**2016**
logstic regression on 1-3 grams baseline	0.627	0.629	0.586	0.558
CNN(word2vec ,convolution size=[3,4,5])	0.715	0.723	**0.688**	0.643
CNN(fasttcxt,convolution size=[3,4,5])	0.720	0.733	0.665	0.640
CNN(glove ,convolution size=[3,4,5])	0.709	0.714	0.660	0.637
CNN(word2vec ,convolution size=[1,2,3])	0.712	0.735	0.673	0.642
CNN(word2vec ,convolution size=[5,6,7])	0.710	0.732	0.676	0.646
CNN(word2vec ,convolution size=[3,4,5]),no class weights	682	0.679	0.659	0.640
CNN(word2vec ,convolution size=[3,4,5]),no distant training	0.698	0.716	0.660	0.636
CNN(word2vec ,convolution size=[3,4,5]),no fully connected layer	0.715	0.724	0.683	0.641
LSTM(word2vec)	720	0.733	0.677	0.636
LSTM(fasttext)	0.712	0.730	0.666	0.633
LSTM(glove)	0.710	0.730	0.658	0.630
LSTM(word2vec,no class weights)	0.689	0.661	0.652	0.643
LSTM (word2vec,no distant training)	0.698	0.719	0.647	0.629
LSTM (word2vec,no fully connected layer)	0.719	0.725	0.675	0.634
Ensemble model	0.725	**0.748**	0.679	**0.648**
previous best histcrical scores	**0.728**	0.744	0.671	0.633

Table II.1: validation results on histcrical test set subtask A

Table II.2 shows Results for Subtask A 2017 ,English [35]:

	System	Recall	F1-score	Accuracy
1	DataStories	0.681	0.677	0.651
2	BB-twtr	0.681	0.685	0.658
3	lia	0.676	0.674	0.661
4	Senti17	0.674	0.665	0.652
5	NNEMBs	0.669	0.0.658	0.664
6	Tweester	0.659	0.648	0.648
7	NGEOTEC	0.649	0.645	0.633
8	SiTAKA	0.645	0.628	0.643
9	TSA-INF	0.643	0.620	0.616
10	USC-NLP	0.642	0.624	0.565
11	HLP@UËNN	0.637	0.632	0.646
12	YNU-HPCC	0.633	0.612	0.647
13	SENTIME++	0.633	0.613	0.601
14	ELIRF-UPV	0.632	0.619	0.599
15	ECNU	0.628	0.613	0.630
16	TAKELAB	0.627	0.607	0.628
17	DUTH	0.621	0.605	0.640
18	GRYSTALNEST	0.619	0.593	0.629
19	DEEPSA	0.618	0.587	0.616
20	NLC-USP	0.612	0.595	0.617
21	TI-SENTI	0.607	0.577	0.627
22	BUSEM	0.605	0.587	0.603
23	EICA	0.595	0.555	0.599
24	OMAM	0.590	0.542	0.615
25	ADULLAN	0.589	0.552	0.614
26	NILETMRG	0.578	0.515	0.606
27	AMOBEE-C-137	0.575	0.520	0.587
28	EJ-ZA-2017	0.571	0.539	0.582
29	LSIS	0.571	0.561	0.521
30	XJSA	0.556	0.519	0.575
31	NEVERLAND-THU	0.555	0.507	0.597
32	MI-T-lab	0.551	0.522	0.561
33	DEGOREF	0.546	0.527	0.540
34	XIWU	0.479	0.365	0.547
35	SSN-MLRGI	0.431	0.344	0.439
36	YNUDLG	0.340	0.201	0.387
37	WARWICKDCS	0.335	0.221	0.382
38	AVID	0.335	0.163	0.206
B1	ALL POSITIVE	0.333	0.161	0.193
B2	ALL NEGATIVE	0.333	0.244	0.323
B3	ALL NEUTRAL	0.333	0.000	0.483

Table II.2: Results for Subtask A 2017 ,English

II.3 Conclusion

In conclusion the application of various machine learning models,has significantly enhanced the efficiency and accuracy of sentiment analysis and opinion mining. These models have been instrumental in handling large datasets and providing more reliable insights into public sentiment. As the field continues to evolve, further advancements and the integration of new techniques are expected to refine the process, offering even more precise and nuanced understanding of sentiments across different domains.

CHAPTER III

Proposed Methods

III.1 Introduction

In this chapter, we detail the methods proposed for our study on Sentiment Analysis or Opinion Mining. We begin by describing the data used in our experiments, followed by the steps taken for data preparation and data cleaning . This chapter outlines the various techniques and approaches considered, including the rationale behind selecting each method.We will cover the algorithms and vectorization techniques employed in our experiments, providing a thorough explanation of their implementation. Additionally, we provide a visual representation of our approach, illustrating the procedural flow and interrelation of our methodological steps, which offers a quick overview of our study methodology as shown in FigIII.1.

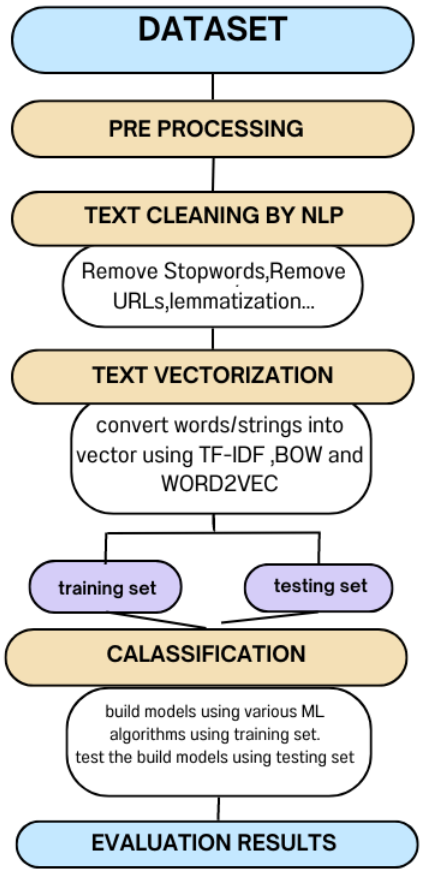

Figure III.1: a quick overview of our study methodology

III.2 Data

SemEval: This stands for the International Workshop on Semantic Evaluation [36], a forum where researchers evaluate systems for tasks related to understanding the meaning of text.

Twitter Data: This refers to a collection of tweets, which are short messages posted on the social media platform Twitter.

In the context of SemEval, Twitter data is used for a specific competition or ”task” related to sentiment analysis.

Here are some variations of this data:

Sentiment labels: Tweets might be labeled as positive, negative, or neutral to train systems that can automatically determine the sentiment of a tweet.

Topic sentiment: Tweets might be labelled with sentiment towards a specific topic, like a product or event.

Sentiment intensity: Data might indicate not just positive or negative, but the strength of that sentiment.

III.2.1 SemEval Data and tasks

- **SemEval-2013 Task 5: Sentiment Analysis in Twitter**

  This task deals with sentiment labels from 2013.

  Labels: Tweets were labeled as positive, negative, or neutral. This SemEval task focused on building systems that could automatically classify the sentiment of a tweet based on these three categories.

- **SemEval 2014: Sentiment Analysis in Twitter:**

  SemEval 2014 included a few tasks that utilized Twitter data for sentiment analysis, but not all focused solely on sentiment labels like the 2013 task. Here's a breakdown of the relevant tasks:

  Task 4: Aspect Based Sentiment Analysis (ABSA)

  This task focused on identifying aspects of products or services mentioned in tweets and determining the sentiment towards those aspects. For example, in a tweet "Terrible battery life on the new phone disappointed," the system would need to identify "battery life" as the aspect and "negative" sentiment.

  Task 9: Sentiment Analysis in Twitter Focusing on Aspects and Sentiments. Similar to Task 4, this task involved identifying aspects and sentiment. However, it might have included additional complexities like sentiment intensity (very negative, slightly positive) or multiple aspects within a single tweet.

- **SemEval 2015:**

  While SemEval 2015 offered various tasks, none directly focused solely on sentiment labels for Twitter data like SemEval 2013. Here's a breakdown of the relevant tracks:

  Task 1:Paraphrase and Semantic Similarity in Twitter (PIT) This task didn't involve sentiment labels. It focused on: Identifying if two tweets convey similar meanings (paraphrase). Measuring the degree of semantic similarity between tweets.

  Task 12: Aspect Based Sentiment Analysis (ABSA) Similar to SemEval 2014 tasks, ABSA aimed to:

Identify aspects of entities (products, services, etc.) mentioned in text (not just Twitter). Determine the sentiment towards those aspects.

- **SemEval 2016:** SemEval 2016 included a sentiment analysis task that used Twitter data! Here's the relevant task:

  Task 4: Sentiment Analysis in Twitter

  This task revisited the sentiment analysis focus from SemEval 2013, but with some potential variations:

  The sentiment labels might have remained positive, negative, and neutral, but there's a possibility they included more granular labels or focused on sentiment intensity (e.g., very negative, slightly positive).

- **SemEval 2017:**

  SemEval 2017 offered a task that used Twitter data for sentiment analysis, but it also included some humor and truth detection aspects. Here's the relevant task:

  Task 4: Sentiment Analysis in Twitter

  This task revisited sentiment analysis, but with some potential variations from previous years:

  Sentiment Labels: The labels might have included positive, negative, and neutral, or they could have explored sentiment intensity or targeted sentiment towards specific topics.

  Humor and Truth Detection: The task might have incorporated elements of identifying humor (e.g., sarcasm) within tweets and potentially detecting the truthfulness of the sentiment expressed.

III.2.2 Dataset

SemEval-2017 Task 4 consists of five subtasks, each offered for both Arabic and English:

- **Subtask A: Message-Level Sentiment Classification:** Classify the overall sentiment of a tweet (e.g., positive, negative, neutral).

- **Subtask B: Topic-Level Sentiment Classification (2-point scale):** Classify the sentiment of a tweet towards a specific topic using a binary scale (e.g., positive, negative).

- **Subtask C: Topic-Level Sentiment Classification (5-point scale):** Similar to Subtask B, but using a more granular 5-point scale for sentiment (e.g., very negative, slightly positive).

- **Subtask D: Message-Level Sentiment Distribution:** Quantify the distribution of sentiment across a set of tweets (e.g., percentage of positive, negative, and neutral tweets).

- **Subtask E: Topic-Level Sentiment Distribution (2-point scale):** Similar to Subtask D, but focusing on sentiment distribution towards a specific topic using a binary scale.

- **Subtask E': Topic-Level Sentiment Distribution (5-point scale):** Similar to Subtask D, but focusing on sentiment distribution towards a specific topic using a 5-point scale.

In this work, we use the SemEval 2017 Task 4, Subtask A dataset as our primary source of data. SemEval (Semantic Evaluation) provides a valuable resource for natural language processing research, offering diverse tasks that challenge the boundaries of text understanding and processing. Specifically, we have chosen Subtask A of SemEval 2017 Task 4 due to its relevance and significance in the field. Our purpose in selecting this dataset is to explore and analyze the performance of various machine learning and natural language processing models in addressing sentiment analysis tasks, leveraging the rich annotations and complexities present within the SemEval data.

Dataset	Positive	Neutral	Negative	Total
twitter-2013train-A	3640	4586	1458	9684
twitter-2013test-A	1475	1513	559	3547
twitter-2013dev-A	575	739	340	1654
twitter-2014sarcasm-A	20	7	22	49
twitter-2014test-A	982	669	202	1853
twitter-2015test-A	1038	987	365	2390
twitter-2015train-A	170	253	66	489
twitter-2016dev-A	829	746	391	1966
twitter-2016devtest-A	994	681	325	2000
twitter-2016test-A	170	253	66	489
twitter-2016train-A	3017	2001	850	5868

Table III.1: SemEval 2017 Task 4 Subtask A :Train Dataset

```
1 628949369883000832  negative  dear @Microsoft the newOoffice for Mac is great and all, but no Lync update? C'mon.
2 628976607420645377  negative  @Microsoft how about you make a system that doesn't eat my friggin discs. This is the 2nd time th
3 629023169169518592  negative  I may be ignorant on this issue but... should we celebrate @Microsoft's parental leave changes? D
4 629179223232479232  negative  Thanks to @microsoft, I just may be switching over to @apple.
5 629186282179153920  neutral If I make a game as a #windows10 Universal App. Will #xboxone owners be able to download and play t
6 629226490152914944  positive  Microsoft, I may not prefer your gaming branch of business. But, you do make a damn fine operatin
7 629345637155360768  negative  @MikeWolf1980 @Microsoft I will be downgrading and let #Windows10 be out for almost the 1st yr ba
8 629394528336637953  negative  @Microsoft 2nd computer with same error!!! #Windows10fail Guess we will shelve this until SP1! ht
9 629650766580609026  positive  Just ordered my 1st ever tablet; @Microsoft Surface Pro 3, i7/8GB 512GB SSD. Hopefully it works c
10 629797991826722816  negative  After attempting a reinstall, it still bricks, says, "Windows cannot finish installing," or some
11 630159517058142208  positive  Sunday morning, quiet day so time to welcome in #Windows10 @Microsoft @Windows http://t.co/7VtvA
12 630542330827771904  negative  Did @Microsoft break Windows 10? Was working fine on Wednesday but now I can't get passed the log
13 630636736746422272  negative  @MSAU @Microsoft spent over 40 mins &amp;amp; gave up coz it changes just before you print every time
14 630807124872970240  neutral @spyderharrison @Microsoft the reason I ask is because it may be the manufacturer's fault, and they
15 630818265799921664  positive  Innovation for jobs is just around the corner - to be exact next Wednesday 8/19 at @Microsoft htt
16 630909171437801472  neutral OK this is my pure speculation.  @Microsoft owns the cloud compute tech.  @Cloudgine is utilizing t
17 630982270409572352  neutral We are still taking registrations for our Education Technology Update with @Acer and @Microsoft on
18 631104156187627520  negative  For the 1st time @Skype has a "High Startup impact"   Does anyone at @Microsoft have a clue? #Win
19 631223085476261890  negative  #teens @BillGates 1st company failed miserably. When Gates &amp;amp; @PaulGAllen tried to sell the pr
20 631368262979297281  positive  #Vote for @AIESEC to become the 10th Global non profit partner of @Microsoft for us to #UpgradeYo
21 631521079245307904  positive  Top 5 most searched for Back-to-School topics -- the list may surprise you http://t.co/Xj21uMVo0p
22 631543121407442946  negative  @Microsoft support for 365 has been terrible. On the phone for an hour then got dropped. So far n
23 631696872323850240  positive  @trucker_squigz @Microsoft @MISpeedway @nationwide88 Cant wait to see you up here. I will be at t
24 631792365590695936  negative  Again-just like #Windows 8- @Microsoft makes 10 w/o the start menu their best customers want - 3r
25 631842974268305408  positive  @ScottArbeit @GabeAul @Microsoft isntall the newest version and you may chance your mind!
```

Figure III.2: Samples from SemEval 2017 Task 4 Subtask A:train Dataset

III.2.3 Data Preparation

The dataset should be labelled and pre-processed before giving to ML models. we have used the simple code(as shown in figure III.3) to label the data into positive , negative and neutral based on the polarity generated for the tweet.

```python
def transform_labels(label):
    if label=='positive':
        return 2
    if label=='negative':
       return 0
    if label=='neutral':
       return 1
    return label

df['Sentiment'] = df['Sentiment'].apply(lambda x:transform_labels(x))
```

Figure III.3: data labeling code snippet

III.2.3.1 Feature Engineering: Word Count as a Predictor in Sentiment Analysis

In our sentiment analysis endeavor, we recognize the importance of feature engineering to extract meaningful insights from textual data. One such feature that proves invaluable is word count. By quantifying the number of words within each text passage, we gain a nuanced understanding of its complexity and length, factors that can significantly influence sentiment interpretation.

III.2.4 Training and Test sets

The dataset are randomized split into two distinct subsets: training and testing, with a distribution ratio of 80:20, respectively. Maintaining class balance across subsets is crucial, achieved by leveraging the stratify parameter within the scikit-learn function train-test-split(). The model is trained on the training subset, while the testing subset evaluates the model's performance.

III.2.5 Data Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. All datasets were cleaned using regular expression[1] via the following steps:

III.2.5.1 Remove Twitter Handles

Any Twitter handles (strings starting with '@') are removed from the text.

[1] regular expression is a sequence of characters used to match strings of text such as particular characters, words, or patterns of characters.

III.2.5.2 Remove URLs

URLs can introduce noise into the dataset. They may contain random characters, symbols that are not relevant to the analysis. By removing it, we reduce the amount of noise in the text data.

III.2.5.3 Remove Punctuation, Numbers and Special Characters

Any characters that are not letters or the '' symbol are removed from the text.

III.2.5.4 Lower casing

Lowercase conversion of a word (WORD $\rightarrow$ word). Name and name have the same meaning, but when not changed to lower case, they are treated as separate words in the vector space model (resulting in more dimensions).

III.2.5.5 Remove Stopwords

Common English stopwords, like "and", "the", "is", etc., are removed from the text. However, the word " **not**" is retained, likely due to its importance in sentiment analysis.

III.2.5.6 Stemming

is the process of removing the last few characters of a given word, to obtain a shorter form (Finally , final $\rightarrow$ Fina) using the Porter stemming algorithm, even if that form doesn't have any meaning.is used to get that base word.

III.2.5.7 lemmatization

reducing inflected words to their root forms, known as lemmas. For instance, the lemma of "running" is "run," and that of "better" is "good." This process involves algorithmic identification of an inflected word's lemma, which represents its dictionary form and encapsulates its intended meaning.

```python
import nltk
stopword=nltk.corpus.stopwords.words('english')
stopword.remove('not')
pt=PorterStemmer()
wordnet=WordNetLemmatizer()
for index,text_ in enumerate(text):
    text_=re.sub(r'@[\w]*','',text_) #Removing Twitter Handles (@user)
    text_=re.sub(r'http/S+','',text_) #Removing urls from text
    text_=re.sub(r'[^A-Za-z#]',' ',text_) #Removing Punctuations, Numbers, and Special Characters
    text_=" ".join(i.lower() for i in text_.split() if i.lower() not in stopword) #Removing stopword
    text_=" ".join(pt.stem(i) for i in text_.split()) #Stemming the word
    text_=" ".join(wordnet.lemmatize(i) for i in text_.split())
    text[index]=text_
```

Figure III.4: Text processing code snippet

	Tweet_ID	Sentiment	Tweet_Text	wordcount	Clean_Tweet	length_tweet
0	264183816548130816	positive	Gas by my house hit $3.39!!!! I\u2019m going t...	14	ga hous hit u go chapel hill sat	32
1	263405084770172928	negative	Theo Walcott is still shit\u002c watch Rafa an...	14	theo walcott still shit u c watch rafa johnni ...	59
2	262163168678248449	negative	its not that I\u2019m a GSP fan\u002c i just h...	16	not u gsp fan u c hate nick diaz u wait februari	48
3	264249301910310912	negative	Iranian general says Israel\u2019s Iron Dome c...	22	iranian gener say israel u iron dome u deal mi...	78
4	262682041215234048	neutral	Tehran\u002c Mon Amour: Obama Tried to Establi...	21	tehran u c mon amour obama tri establish tie m...	102

Figure III.5: Data Cleaning

III.2.6 Bag of Words Model

The Bag of Words [37] model is one of the most straightforward yet effective methods for extracting features from text documents. This model's core principle involves transforming text documents into vectors, where each document is represented by a vector indicating the frequency of every distinct word within the document's vector space. Therefore, for our sample vector from the previous mathematical notation for D, the weight of each word corresponds to its frequency in that document .

III.2.7 TF-IDF Model

TF-IDF [38] which stands for Term Frequency-Inverse Document Frequency, combines two metrics: term frequency and inverse document frequency. Initially developed as a metric for ranking search engine results based on user queries, it is now integral to information retrieval and text feature extraction. Let's formally define TF-IDF and examine its mathematical representation.

TF-IDF is the product of two metrics, expressed as:

$$tfidf = tf \times idf \tag{III.1}$$

Here, term frequency (tf) and inverse document frequency (idf) are the two metrics involved. Term frequency (tf) is what we calculated in the Bag of Words model. It represents the raw frequency value of a term in a particular document and is mathematically defined as:

$$tf(w, D) = f_{w_D} \tag{III.2}$$

where f_{w_D} denotes the frequency of word W in document D.
Inverse document frequency (idf) is the inverse of the document frequency for each term. It is calculated by dividing the total number of documents in the corpus by the document frequency for each term and then applying logarithmic scaling: .

$$idf(t) = 1 + \log \frac{C}{1 + df(t)} \tag{III.3}$$

where idf(t) represents the idf for term t, C is the total number of documents in the corpus, and df(t) is the number of documents containing term t .

III.2.8 Word2Vec

Word2vec is a technique used in natural language processing (NLP) to turn words into numerical representations. These representations, called word embeddings, capture the meaning of a word based on the words around it in a large text corpus. By analyzing the context of a word, word2vec can map words with similar meanings to similar locations in a vector space. This allows us to perform operations on words that reflect their semantic relationships. Word2vec itself is not a single algorithm, but rather a family of related models. These models are typically shallow neural networks that are trained to predict the surrounding words of a given word. By analyzing how well the model predicts these surrounding words, we can update the word embeddings to better capture the word's meaning.
Here are the key aspects that guide this process:

1. **Training Algorithm:** There are two main algorithms used to train the word vectors:

- **Negative Sampling:** This is the more common method today. This method involves calculating a probability of a word appearing in a specific context. Here's a simplified breakdown:

  Let's say we have a word w and a context word c.

  We want to estimate the probability P(c | w) how likely c appears after w in the training corpus.Word2vec uses a sigmoid function (represented as) to model this probability:

  $$\sigma(t) = \frac{1}{1+e^{-t}}$$

  The function takes a dot product of the word vector vw for w and the context word vector vc for c as input:

  $$t = vw^T \times vc$$

2. **Subsampling:** Extremely frequent words (like "the") and very rare words don't provide much semantic information. Word2vec downweights the influence of these words during training, focusing on more informative vocabulary.

3. **Dimensionality and Context Window:** These are hyperparameters you tune during training. The number of dimensions in the vector space determines the complexity of the relationships captured between words. A larger window size for considering surrounding words provides more context but can also introduce noise.

4. **Training Objective:** The core idea is to predict surrounding words (CBOW) or the central word (Skip-gram) based on the current word vectors. The model calculates the difference between the predicted and actual words (error).

5. **Backpropagation:** The word vectors are then updated to minimize this error through a technique called backpropagation. This refines the vectors with each iteration.

6. **Iteration and Convergence:** The training process goes through the text corpus multiple times, updating word vectors based on prediction errors. This continues until the model converges, meaning the word vectors no longer change significantly.

III.3 Implementation

Depending on the data set, a specific processing approach was then used to facilitate it Building the Model Using the Example Twitter Dataset, we named the columns because they were not named before. This way: Tweet-ID, Sentiment, Tweet-Text .

Then, we process the tweet by apply the steps we mentioned before.

After process and cleaning the datasets, sentences were tokenized into individual words and then lemmatized for standardization. Following this, the Bag of Words (BoW) vocabulary, TF-IDF vocabulary, and Word2Vec embeddings were constructed. BoW represents word frequencies, TF-IDF represents word importance, and Word2Vec captures semantic relationships between words.These constructed vocabularies and embeddings, were then used as inputs for various machine learning models including Naive Bayes, Random Forest, Decision Trees, and Linear Support Vector Machines (LSVM). Each model was trained to perform sentiment analysis using the respective vocabularies and embeddings.

After training, the models were evaluated using appropriate metrics to assess their performance. Commonly used metrics such as accuracy and F1-score were calculated. This comprehensive evaluation allowed for a thorough understanding of each model's performance, aiding in the selection of the best-performing model .

III.4 Conclusion

In conclusion, this chapter has provided a detailed overview of the methodologies employed in our study on Sentiment Analysis. We began by The selection and preparation of data and datasets were thoroughly discussed, ensuring a robust foundation for our analysis. Various text vectorization methods were examined to effectively transform textual data into a suitable format for our algorithms.Finally, we outlined the steps of our final implementation, integrating these components into a cohesive framework.

CHAPTER IV

Results and Conclusion

IV.1 Introduction

In this chapter, we will present a comprehensive overview of our study, starting with the software tools utilized during our research. Following this, we will present the results obtained from our experiments in a clear and organized manner. Our goal is to provide an in-depth analysis of the collected data, highlighting key findings and insights. We will systematically explore the outcomes of each experiment, discussing their implications and relevance to our study. Furthermore, we will compare our results with those from other related works to contextualize our findings within the broader field of research.

IV.2 Software Tools

IV.2.1 Numpy

NumPy (an abbreviation for Numerical Python) offers a powerful way to store and manipulate dense data buffers. Although similar to Python's built-in *list* type, NumPy arrays are far more efficient for storage and data operations as they increase in size.[39].

IV.2.2 Pandas

Pandas is a more recent library developed on top of NumPy, offering an efficient implementation of a *DataFrame*. *DataFrames* are essentially multidimensional arrays with row and column

labels, frequently containing various data types and/or missing values. In addition to providing an easy-to-use storage interface for labeled data, Pandas includes numerous powerful data operations that will be familiar to users of database frameworks and spreadsheet software.[39].

IV.2.3 NLTK

The Natural Language Toolkit (NLTK) is a comprehensive platform that includes over 50 corpora and lexical resources. It also offers essential tools, interfaces, and methods for processing and analyzing text data. The NLTK framework features a collection of efficient modules for tasks such as classification, tokenization, stemming, lemmatization, tagging, parsing, and semantic reasoning. It is the industry-standard tool for any NLP project.[40]

IV.2.4 Sklearn

Several Python libraries offer robust implementations of various machine learning algorithms. One of the most renowned is Scikit-Learn, a package that delivers efficient versions of numerous common algorithms. Scikit-Learn is known for its clean, consistent, and streamlined API, along with its highly useful and comprehensive documentation. This consistency means that once you grasp the basic usage and syntax of Scikit-Learn for one model, transitioning to a different model or algorithm is simple [39].

IV.2.5 matplotlib

Matplotlib is a comprehensive Python library used for creating static, animated, and interactive visualizations. It provides an extensive range of plotting functions and tools to generate high-quality graphs, charts, and plots. Matplotlib's versatility allows users to customize every aspect of a figure, from line styles and colors to axes properties and annotations. It is widely used in data analysis and scientific research to produce clear and informative visual representations of data,

IV.2.6 seaborn

Seaborn is a Python visualization library built on top of Matplotlib that provides a high-level interface for creating attractive and informative statistical graphics. It simplifies complex plotting functions and enhances the aesthetics of graphs with themes, color palettes, and built-in support for handling data frames from libraries like Pandas. Seaborn is particularly well-suited for visualizing data relationships and trends, making it a popular choice for data analysis and exploration.

IV.2.7 Gensim

Gensim is a Python library designed for topic modeling, document indexing, and similarity retrieval across large text collections. It is aimed at users working in the field of natural language processing (NLP)[41] .

IV.3 Experiment 1 - Results

This experiment is conducted on the dataset 2013,2014,2015,2016 . Table IV.3 shows the experimental results obtained by applying different machine learning classifiers, namely NB, LR, LinearSVC,KNN, DT, RF, in order to evaluate the effectiveness of those methods while using feature extraction methods including TF-IDF,BOW as well as Word2Vec and also in terms of different evaluation measures like accuracy and f1-score.

	MODEL	2013		2014		2015		2016	
		F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy
BOW	BernoulliNB	0.59	0.62	0.52	0.59	0.54	0.59	0.60	0.60
	MultinomialNB	0.62	0.63	0.58	0.61	**0.59**	**0.60**	0.60	0.61
	LogisticRegression	**0.66**	**0.68**	**0.61**	**0.63**	0.57	0.59	**0.62**	**0.62**
	LinearSVC	0.62	0.63	**0.61**	0.62	**0.59**	**0.60**	0.59	0.59
	RandomForest	0.64	0.67	0.58	0.60	0.56	0.58	0.57	0.59
	DecisionTree	0.63	0.64	0.60	0.60	0.54	0.55	0.54	0.55
	KNN (k=3)	0.44	0.46	0.45	0.45	0.42	0.44	0.44	0.48
	KNN (k=5)	0.44	0.50	0.41	0.41	0.40	0.44	0.45	0.49
	KNN (k=7)	0.43	0.50	0.42	0.43	0.40	0.45	0.44	0.49
	KNN (k=11)	0.40	0.49	0.39	0.41	0.37	0.44	0.41	0.48
TF-IDF	BernoulliNB	0.66	0.66	**0.65**	**0.65**	0.56	0.56	**0.61**	**0.62**
	MultinomialNB	0.63	0.65	0.59	0.63	0.57	**0.59**	**0.61**	0.61
	LogisticRegression	0.63	0.66	0.57	0.63	0.51	0.56	**0.61**	**0.62**
	LinearSVC	**0.67**	**0.67**	0.64	0.64	**0.59**	**0.59**	**0.61**	0.61
	RandomForest	0.65	**0.67**	0.59	0.60	0.55	0.56	0.58	0.59
	DecisionTree	0.58	0.58	0.60	0.60	0.52	0.53	0.52	0.52
	KNN (k=3)	0.50	0.50	0.49	0.48	0.49	0.49	0.35	0.45
	KNN (k=5)	0.52	0.53	0.49	0.49	0.51	0.51	0.35	0.45
	KNN (k=7)	0.53	0.54	0.54	0.54	0.52	0.54	0.37	0.47
	KNN (k=11)	0.54	0.56	0.54	0.54	0.51	0.53	0.43	0.50
Word2Vec	BernoulliNB	0.53	0.52	0.52	0.50	0.48	0.48	0.53	0.53
	LogisticRegression	**0.61**	**0.62**	0.56	0.57	**0.55**	**0.56**	**0.58**	**0.58**
	LinearSVC	0.60	**0.62**	**0.58**	**0.59**	0.54	**0.56**	**0.58**	**0.58**
	RandomForest	0.56	0.59	0.55	0.57	0.52	0.55	**0.58**	**0.58**
	DecisionTree	0.55	0.55	0.47	0.477	0.46	0.46	0.46	0.46
	KNN (k=3)	0.47	0.46	0.44	0.43	0.46	0.46	0.49	0.49
	KNN (k=5)	0.50	0.50	0.45	0.45	0.46	0.46	0.51	0.51
	KNN (k=7)	0.51	0.51	0.45	0.46	0.47	0.47	0.52	0.52
	KNN (k=11)	0.50	0.51	0.50	0.52	0.47	0.48	0.53	0.53

Table IV.1: Comparative results of different machine learning classifiers on different dataset

IV.4 Experiment 1 - Analysis

- **WORD2VEC :**

1. **2013 Dataset :**

   Top Performers: Logistic Regression and LinearSVC outperformed in terms of accuracy (0.62) for both , followed by RandomForest by achieving accuracy (0.59).

   Consistency: Logistic Regression and LinearSVC demonstrate consistent high per-

formance across both accuracy and F1-score.

Decision Tree: Performs relatively well but falls slightly in comparison TO other classi fiers in terms both of the applied evaluation metrics, by achieving only 0.55

2. **2014 Dataset :**

   Top Performers: LinearSVC outperformed in terms of in terms of accuracy (0.59)and F1-score(0.58),followed closely by Logistic Regression and RandomForest.

   Consistency: LinearSVC demonstrates consistent high performance across both metrics, also Logistic Regression and RandomForest showing consistent results, albeit slightly lower.

   Decision Tree has again the less performer in term of the applied evaluation metrics in comparison to other models,where both of F1-score and accuracy value it reached was 0.47

3. **2015 Dataset :**

   Top Performers: Logistic Regression and LinearSVC models achieve the best accuracy scores of 0.56, with Logistic Regression also having the best F1-score of 0.55.

   Consistency: Logistic Regression, LinearSVC, and RandomForest demonstrate consistent performance, with matching accuracy scores. Decision Tree performs the poorest among all models.

4. **2016 Dataset :**

   Top Performers: The Logistic Regression, LinearSVC, and Random Forest models stand out as the top performers in terms of both the applied evaluation metrics, by achieving in terms of accuracy (58)and F1-score(58%).

   Consistency: These top-performing models demonstrate consistent high performance across both accuracy and F1-score.

   Decision Tree: While the Decision Tree model achieves a lower accuracy score of 0.46 and an F1-score of 0.46, it shows potential for improvement.

   KNN and Naive Bayes Models: KNN and BernoulliNB perform relatively poorly compared to the top-performing models, with accuracy and F1-score between 0.49 and 0.53 .

- **TF-IDF:**

1. **2013 Dataset :**

   both of LinearSVC, and RandomForest outperformed other classifiers in terms of accuracy by achieving 0.67 for both.

   Consistency: LinearSVC and RandomForest including Logistic Regression demonstrate consistent performance with high accuracy and F1-score.

   Decision Tree: Performs relatively poorly in comparison to other models, with accuracy and F1-score both at 0.58.

2. **2014 Dataset :**

   Top Performers: BernoulliNB achieves the best accuracy and F1-score(both at 0.65), closely followed by LinearSVC and Logistic Regression.

   Consistency: LinearSVC demonstrates consistent performance with an accuracy score and F1-score both at 0.64, followed by Logistic Regression with consistent scores (0.63 accuracy and 0.57 F1-score).

   Decision Tree: Performs reasonably well but slightly lower than the top models, with an accuracy score and F1-score both at 0.60.

3. **2015 Dataset :**

   Top Performers: MultinomialNB and LinearSVC outperformed other classifiers in terms of accuracy by achieving 0.59 for both

   . **Consistency:** Both MultinomialNB and LinearSVC demonstrate consistent performance with accuracy and F1-score.

   Decision Tree: Performs relatively poorer compared to other models, with accuracy and F1-score at 0.53 and 0.52 respectively.

4. **2016 Dataset :**

   Top Performers: BernoulliNB, Logistic Regression, and MultinomialNB achieve similar F1-score and accuracy results between 0.60 and 0.62 .

   Consistency: LinearSVC demonstrates consistent performance with both accuracy and F1-score at 0.61.

   KNN with K=3 and K=5 using has the worst performer in terms of F1-score (0.35) in comparison to WORD2VEC and BOW

- **BOW :**

1. **2013 Dataset :**

   Top Performers: Logistic Regression outperformed in terms of accuracy (0.68) and F1-score(0.66), closely followed by RandomForest with an accuracy of 0.67 and an F1-score of 0.64

   Consistency: RandomForest demonstrates consistent performance with both accuracy and F1-score at 0.67 and 0.64 respectively.
   LinearSVC and DecisionTree also show relatively consistent performance with both the applied evaluation metrics above 0.62.
   MultinomialNB vs. BernoulliNB:MultinomialNB outperforms BernoulliNB with higher accuracy and F1-score.

2. **2014 Dataset :**
   Top Performers: Logistic Regression outperformed in terms of accuracy 0.63, followed by LinearSVC with an accuracy of 0.62.
   Logistic Regression also outperformed in terms of F1 score (0.61), followed closely by LinearSVC with an F1 score of 0.61.
   Consistency: DecisionTree, RandomForest, and MultinomialNB demonstrate consistent performance with accuracy scores around 0.60 and F1-score around 0.58.
   BernoulliNB was able to equate Top Performersmodels, but failed at 0.52 and 0.59 in accuracy and F1-score respectively.

3. **2015 Dataset :**
   Top Performers: both of LinearSVC and MultinomialNB achieves the best accuracy score of 0.60 and an F1-score of 0.59 for both .
   BernoulliNB and Logistic Regression were able to equate Top Performers models, but failed with accuracy and F1scores with values that were between 0.57 and 0.59 for both.
   Consistency: MultinomialNB, LinearSVC, and RandomForest demonstrate consistent performance with accuracy scores around 0.60 and F1-score around 0.59.

DecisionTree performs relatively poorly compared to other models, with accuracy and F1-score both at 0.55 and 0.54 respectively.

4. **2016 Dataset :**

   Top Performers: Logistic Regression achieves the best accuracy and F1-score, both at 0.62, indicating strong performance.

   MultinomialNB closely follows with an accuracy of 0.61 and an F1 score of 0.60.

   Consistency: RandomForest, LinearSVC, and MultinomialNB demonstrate consistent performance with accuracy scores around 0.59 and F1 F1-score between 0.57 and 0.60.

   DecisionTree performs relatively Linearpoorly compared to other models, with accuracy and F1-score both at 0.55 and 0.54 respectively.

IV.5 Observations

Based on previous experiments and results, we highlight some observations:

- Bag-of-Words (BoW) achieves similar to TF-IDF performance , especially when used with Logistic Regression, LinearSVC, and Random Forest classifiers.
- TF-IDF representation seems to provide the best overall performance, with a significantly smaller vocabulary (2981 words) Compared to bow vocabulary (31889 words).
- A smaller vocabulary could potentially lead to loss of information but it might also result in faster trainig and inference times,especially if the classifier can effectively learn from this reduced vocabulary
- Word2Vec which uses word embedding less outperfomce because the limited amount of data and insufficient context for capture semantic relationships between words.

IV.6 Dicussion

In this section, we discuss the results obtained from our study on Sentiment Analysis or Opinion Mining.After conducting various experiments and analyses, The Experiment and the results presented in IV.3 shown that the TF-IDF method was the most effective vectorization technique. Additionally, the LinearSVC model demonstrated the best performance when paired with TF-IDF vectorization .

We explore the implications of these findings, highlighting the strengths and weaknesses of the various approaches used. Furthermore, we address any challenges encountered during the study and consider their impact on the overall results. Future research may investigate how these and other strategies might be improved to provide even better outcomes.
This discussion aims to provide a comprehensive understanding of our study's outcomes and it's relevance to the field of Opinion Mining.

IV.7 Comparaison

In this work, we conduct a comparative analysis of deep learning and traditional machine learning techniques reported in Chapter II Compared with our experimental results, with the aim of highlighting the strengths and limitations of each approach in addressing the specified problem domain.

The Best results	**System**	**F1-Score**			
		2013	**2015**	**2014**	**2016**
Deep learning	ensemble model	0.725	0.748	0.679	0.648
Machine learning	logstic regression on 1-3 grams baseline	0.627	0.629	0.586	0.558
Our approach	LinearSVC (TF-IDF)	0.67	0.64	0.59	0.61

Table IV.2: our approach in Comparison to previous works

We compared our results with related work in the field of machine learning and found that our approach outperformed existing methods. However, when comparing our results with related work in deep learning, we found that their approaches achieved better results and outperformed our approaches,so the Deep Learning is the suitable approach for sentiment analysis (opinion mining).

IV.8 Conclusion

In this work, we presented the fundamental machine learning models and associated approaches applied to message polarity classification for the SemEval-2017 Task 4 on Sentiment Analysis in Twitter (subtask A, English). Our goal was to experiment with machine learning models along with modern training strategies in an effort to build the best possible sentiment classifier for tweets. starting with data preparation, including cleaning the dataset by removing stopwords and performing word lemmatization, we applied key sentiment analysis procedures. We transformed the input data using TF-IDF, Bag of Words (BOW) and word embeddings, and used these as inputs for various machine learning models. We conducted a series of experiments testing Naive Bayes, Linear SVC, Random Forest, and other models on the datasets. Additionally, we reviewed relevant studies and conducting our own experiments, we gained a comprehensive understanding of how machine learning models can be applied to sentiment analysis. Our machine learning model achieved good performance, demonstrating the potential of this approach. However, reviewing relevant studies also revealed the sophistication achieved in deep learning models, which our approach's performance didn't quite reach. This highlights the need for further research to improve the accuracy and robustness of machine learning approach in sentiment analysis or opinion mining.

References

[1] Sakshi Indolia, Anil Kumar Goswami, Surya Prakesh Mishra, and Pooja Asopa. Conceptual understanding of convolutional neural network-a deep learning approach. Procedia computer science, 132:679–688, 2018.

[2] Ankit and Nabizath Saleena. An ensemble classification system for twitter sentiment analysis. Procedia Computer Science, 132:937–946, 2018. International Conference on Computational Intelligence and Data Science.

[3] Hao Yang, Chunfeng Yuan, Li Zhang, Yunda Sun, Weiming Hu, and Stephen J Maybank. Sta-cnn: Convolutional spatial-temporal attention learning for action recognition. IEEE Transactions on Image Processing, 29:5783–5793, 2020.

[4] Zhao Jianqiang. Combing semantic and prior polarity features for boosting twitter sentiment analysis using ensemble learning. In 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), pages 709–714, 2016.

[5] Aytuğ Onan, Serdar Korukoğlu, and Hasan Bulut. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Systems with Applications, 62:1–16, 2016.

[6] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up sentiment classification using machine learning techniques. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pages 79–86. Association for Computational Linguistics, July 2002.

[7] Siham Yousfi, Maryem Rhanoui, and Mounia Mikram. Comparative study of cnn and lstm for opinion mining in long text. Journal of Automation, Mobile Robotics and Intelligent Systems, pages 50–55, 2020.

[8] Alpna Patel and Arvind Kumar Tiwari. Sentiment analysis by using recurrent neural network. In Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE), 2019.

[9] Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11, page 129–136, Madison, WI, USA, 2011. Omnipress.

[10] Huaxia Rui, Yizao Liu, and Andrew Whinston. Whose and what chatter matters? the effect of tweets on movie sales. Decision Support Systems, 55(4):863–870, 2013. 1. Social Media Research and Applications 2. Theory and Applications of Social Networks.

[11] Jayashri Khairnar and Mayura Kinikar. Machine learning algorithms for opinion mining and sentiment classification. International Journal of Scientific and Research Publications, 3(6):1–6, 2013.

[12] J Bollen, H Mao, and XJ Zeng. Twitter mood predicts the stock market. journal of computational science. 2: 1–8. 2011.

[13] Tao Xu, Qinke Peng, and Yinzhao Cheng. Identifying the semantic orientation of terms using s-hal for sentiment analysis. Knowledge-Based Systems, 35:279–289, 2012.

[14] Julian Brooke, Milan Tofiloski, and Maite Taboada. Cross-linguistic sentiment analysis: From english to spanish. In Proceedings of the international conference RANLP-2009, pages 50–54, 2009.

[15] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4):1093–1113, 2014.

[16] Xing Liu, Jianfeng Yang, Chengming Zou, Qimei Chen, Xin Yan, Yuao Chen, and Chenran Cai. Collaborative edge computing with fpga-based cnn accelerators for energy-efficient and time-aware face tracking system. IEEE Transactions on Computational Social Systems, 9(1):252–266, 2022.

[17] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. Journal of big Data, 8:1–74, 2021.

[18] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval Vol. 2, Nos. 1–2.

[19] L Lee B Pang. S Vaithyanathan - arXiv preprint cs/0205070.

[20] R Rajasree MS Neethu. Sentiment analysis in twitter using machine learning techniques. 2013 fourth international conference on computing, communications and networking technologies (ICCCNT),, pages 1–5, 2013.

[21] J Ranjan G Dubey, A Rana. Data Analysis..., 2016.

[22] S Ahmad… FM Kundi, A Khan. Scientific Research, 2014.

[23] D Godoy A Tommasel. Information Fusion, 2018 - Elsevier.

[24] Bing Liu. Sentiment Analysis and Opinion Mining. April 22, 2012.

[25] Tobias Daudert. Contextualised sentiment analysis in the financial domain Daudert ,National University of Ireland ,NUI Galway. 2021-07-02.

[26] SM Mohammad S Kiritchenko, X Zhu. Journal of Artificial Intelligence..., 2014.

[27] Philip C. Treleven olga kolchyna, yharsis T.P.Souza and Tomaso Aste. Twitter Sentiment Analysis:Lexicon method,machine learning method and Their Combination.

[28] Procedia computer science. pages Pages 937–946, 2018.

[29] International Journal of Machine Learning and Computing,, 5, October 2018.

[30] S Ahmad… FM Kundi, A Khan. International Journal of Scientific and Research, 6, June 2013.

[31] Mayura Kinikar Jayashri Khairnar. Machine Learning Algorithme for Opinion Minig and Sentiment Classification.

[32] Aditya SharmaA. Daniels. Tweets Sentiment Analysis via Word Embeddings and Machine Learning Techniques Computer Science. ArXiv, 2020.

[33] Sentiment analysis on twitter using machine learning techniques and tf-idf feature extraction: A comparative study. October 2023.

[34] bloomberg mathieu cliche. Bb-twtr at semeval-2017 task4: Twitter sentiment analysus with cnns and lstms.

[35] Preslave Nakov Sara Rosenthal, Noura Farra. Semeval-2017 task4: Sentiment analys.

[36] Semeval:international workshop on semantic evaluation. https://semeval.github.io/.

[37] Sarkar Dipanjan. Text analytics with python: A practical real-world approach to gaining actionable insights from your data, 2016.

[38] seaborn: statistical data visualization. https://seaborn.pydata.org/.

[39] Jake VanderPlas. Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.", 2016.

[40] Dipanjan Sarkar. Text analytics with Python: a practitioner's guide to natural language processing. Springer, 2019.

[41] gensim – topic modelling in python. https://github.com/piskvorky/gensim#documentation.