الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

20 Août 1955 University of Skikda

Faculty of Sciences
Department of Computer Science

Master thesis

For obtaining the diploma of Master degree in Computer Science
Option: Systeme d'Informatique - SI

Subject

Sentiment Analysis of Customer Reviews in Algerian Dialect using DziriBERT: a Transfer Learning Approach

Presented by

REMMACHE Mouhyiddine

and

MEZIADI El Mouatassem Billah

Chairman: Dr.RAMDANE Chafika - 20 Août 1955 University of Skikda.

Reviewer: Mrs.ALI GUECHI Farida - 20 Août 1955 University of Skikda.

Supervisor: Dr.BOUGAMOUZA Fateh - 20 Août 1955 University of Skikda.

– **Session: 2022/2023** –

Dedication

I dedicate this work to myself

To

My dearest father and my dearest mother

To

My elder brother Aymen

To

My sisters Rahma and Ilham

To

My lovly little sister Soundous and nephew Shahine

To

My big family

To

All my friends

.... Mouhyiddine

Dedication

I dedicate this project to Allah Almighty my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength throughout this program.

I also dedicate this modest work to my parents and my sisters, and those who shared with me all the emotional moments during the realization of this work.They warmly supported and encouraged me throughout my journey

.... EL Mouatassem Billah

Acknowledgments

First, we need to thank Allah the Almighty for the good health and well-being that were necessary to complete this work.

الحمد لله

We would like to express our sincere gratitude to our Supervisor Dr.BOUGAMOUZA Fateh the continuous support of us Thesis work, for his advices, patience, motivation, and immense knowledge. His guidance helped us in all the time of research and writing of this thesis.

Besides our Supervisor, we would also like to express our thanks to the members of the jury, who have done us the honor of examining our work.

We thank all the teachers who encouraged and supported us during our studies.

We deeply thank our dear parents, to whom we owe what we are, and our family for supporting us spiritually throughout writing this thesis and ourlife in general.

Abstract

As technology continues to evolve, humans tend to follow suit, and currently social media has taken place as the defacto method of communication. As it tends to happen with verbal communication, people express their opinions in written form and through an analysis of their words, one can extract what an individual wants from a product, a topic, or an event. By looking at the emotions expressed in such content, governments, businesses, and people can learn a lot that can help them improve their strategies.

The information available on the Internet and on social media has become a gold mine for companies developing in their production, services, management and distribution thanks to the millions of comments and tweets published every day.

In this thesis, we propose an approach for deals with the issue for sentiment analysis of a dataset expressed in the Algerian dialect, and formed by the Feedback from customers of Algerian telephone operators (Djezzy, Mobilis, and Ooredoo). To achieve our objectives and based in Transfer Learning, we fine-tune DziriBERT pre-trained model. In order to improve the accuracy of this model, a series of experiments were performed to determine the optimal hyper-parameters, and the best pre-processing on data for this task. The result obtained after the tests is an Accuracy rate equal to 81.88%, which is very encouraging for our case study.

Keywords: Sentiment Analysis, Algerian Dialect, Transfer Learning, pre-processing.

Résumé

Alors que la technologie continue d'évoluer, les humains ont tendance à emboîter le pas, et actuellement les médias sociaux sont devenus la méthode de communication de facto. Comme cela a tendance à se produire avec la communication verbale, les gens expriment leurs opinions par écrit et grâce à une analyse de leurs mots, on peut extraire ce qu'un individu veut d'un produit, d'un sujet ou d'un événement. En examinant les émotions exprimées dans un tel contenu, les gouvernements, les entreprises et les gens peuvent en apprendre beaucoup qui peuvent les aider à améliorer leurs stratégies.

Les informations disponibles sur Internet et sur les réseaux sociaux sont devenues une mine d'or pour les entreprises qui évoluent dans leur production, leurs services, leur gestion et leur diffusion grâce aux millions de commentaires et de tweets publiés chaque jour.

Dans ce mémoire , nous proposons une approche pour traiter le problème de l'analyse des sentiments d'un base de données exprimé en dialecte Algérien, et formé par les Feedbacks des clients des opérateurs téléphoniques Algériens (Djezzy, Mobilis, et Ooredoo). Pour atteindre nos objectifs et basés sur l'apprentissage par transfert, nous affinons le modèle pré-entrainé DziriBERT. Afin d'améliorer la précision de ce modèle, une série d'expérimentations ont été réalisées pour déterminer les hyper-paramètres optimaux, et le meilleur prétraitement sur les données pour cette tâche. Le résultat obtenu après les tests est un taux de précision égal à 81,88%, ce qui est très encourageant pour notre étude de cas.

Mots-clés : analyse des sentiments, dialecte Algérien, l'apprentissage par transfert, prétraitement.

مـلخص

مع استمرار تطور التكنولوجيا، يميل البشر إلى أن يحذوا حذوها، وقد ظهرت وسائل التواصل الاجتماعي حاليًا كطريقة فعلية للتواصل. نظرًا لأنه يحدث غالبًا مع التواصل اللفظي، يعبر الناس عن آرائهم في شكل مكتوب ومن خلال تحليل كلماتهم، يمكن للمرء أن يستخرج ما يريده الفرد من منتج أو موضوع أو حدث. من خلال النظر إلى المشاعر التي يتم التعبير عنها في مثل هذا المحتوى، يمكن للحكومات والشركات والأشخاص تعلم الكثير، مما يمكن أن يساعدهم على تحسين استراتيجياتهم.

لقد أصبحت المعلومات المتوفرة على الإنترنت وعلى وسائل التواصل الاجتماعي منجم ذهب للشركات النامية في إنتاجها وخدماتها وإدارتها وتوزيعها بفضل ملايين التعليقات والتغريدات التي تنشر كل يوم.

في هذه المذكرة، نقترح نهجًا للتعامل مع قضية تحليل المشاعر لمجموعة بيانات معبر عنها باللهجة الجزائرية، والتي تم تشكيلها من خلال التعليقات الواردة من عملاء مشغلي الهاتف الجزائريين (دجيزي وموبيليس وأوريدو). لتحقيق أهدافنا واستناداً إلى التعلم الانتقالي، نقوم بضبط نموذج DziriBERT المُدرَّب مسبقًا. من أجل تحسين دقة هذا النموذج، تم إجراء سلسلة من التجارب لتحديد المعلمات الفائقة المثلى وأفضل معالجة مسبقة للبيانات لهذه المهمة. النتيجة التي تم الحصول عليها بعد الاختبارات هي معدل دقة يساوي 81.88٪، وهو أمر مشجع للغاية بالنسبة لدراسة الحالة الخاصة بنا.

الكلمات المفتاحية: تحليل المشاعر، اللهجة الجزائرية، التعلم الانتقالي، المعالجة المسبقة.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligent
ALGD	Algerian Dialect
ANLP	Arabic Natural Language Processing
ANNs	Artificial Neural Networks
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
DL	Deep Learning
FNN	Feed-Forward Network
LSTM	Long Short-Term Memory
ML	Machine Learning
MSA	Modern Standard Arabic
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NNS	Neural Networks
ReLU	Rectified Linear Units
RNNs	Recurrent Neural Networks
SA	Sentiment Analysis

General Introduction

1. Context of the project

Since the appearance of social networks, millions of people have shared their impressions of their daily lives with their community of friends and network of acquaintances. These communication platforms are also used by companies, researchers and academics interested in people's opinions. However, the active growth of opinions on social networks has led those interested in the field to the impossibility of manual processing of large amounts of data and therefore pushed them to use modern methods with the appearance of artificial intelligence, which has allowed companies and researchers to automate their sentiment analysis processes in real time.

Sentiment analysis is a field that is still evolving with facing several challenges, these challenges are sometimes obstacles to analyze the exact polarity of sentiment since the machine must be trained in order to understand particularities such as emotions, sarcasm, prejudices, understanding nuances, like the human brain does. However, with the right natural language processing techniques these exceptions have been overcome with good solutions that have been answered in many languages and their dialects.

Regarding the Algerian dialect, it did not receive much attention. However, currently there is growing interest from companies and researchers given the increase in the volume of Arabic texts in the social media. It is one of the most identified dialects given the lack of resources and standard treatment compared to other languages and dialects such as linguistic, lexical and syntactic representation compared to the Standard Arabic.

All these aspects make the natural language processing solutions that have been developed for the processing of Arabic languages insufficient in front of such a dialect. However, a lot of research and works have been developed with efficiency to answer the problems of the latter.

In order to carry out these analyzes and works, companies receive large amounts of electronic comments every day from their customers, which are in the form of suggestions, criticisms, and also recommendation. These comments are called "Customer Feedback", they consist of a benefit to companies by identifying the needs of their customers so that they can improve satisfaction and reduce the churn rate.

2. Problem statement

Sentiment analysis is a field in full development due to its many applications and aims to analyze user sentiment in order to have their opinion. However, its application on the comments of the customers of the telephone companies in Algerian dialect as well as the large volume of data obtained each month requires a good tool and technique, which makes it possible to analyze the opinions of their customers effectively.

3. Objectives

In order to propose an innovative solution that responds to the problem posed, we will elaborate a strategy by fine-tune DziriBERT pre-trained model, based on the comments of telephone companie's (Djezzy, Ooredoo, and Mobilis) costomer subscribers on social networks. More precisely, it will be a question of extracting the opinions and comments of internet users expressed in the Algerian dialect on the posts of their company's pages, through various social networks (Facebook, Instagram, etc...), analyzing and classifying them in order to determine the satisfaction or non-satisfaction of customers with the services and products offered by these companies. This general objective can be broken down into the following specific sub-objectives:

- Carry out a pre-processing suitable for the comments with Algerian dialect as well as for the context of the telecommunications domain.
- An experimental study to determine the optimal hyper-parameters, as well as the important effect of these hyper-parameters on the accuracy of the model.

4. Thesis structure

Apart from the introduction and the general conclusion, this thesis is made up of three other chapters organized as follows:

Chapter 1: In this chapter, we will define Natural Language Processing (NLP) and present it's branches, techniques and applications. Moreover, we will delve into sentiment analysis approaches and techniques, offering a clear definition of Sentiment Analysis and its diverse application fields.

Chapter 2: In this chapter, we will present the basic notions of the related domain: Machine Learning (ML), Deep Learning (DL), and description of Neural Networks (NNs).Then we go a little deeper into convolutional neural network (CNN), and recurrent neural networks (RNNs).

Finally, we will explain the Transformer architecture, and present the BERT model used to achieve this work.

Chapter 3: In this last chapter, we will present our approach in detail. We will also describe the experiments of our work and discuss the different results obtained.

Chapter 1 : Natural Language Processing and Sentiment Analysis

1. Introduction

Since machines unable to fully comprehend human natural language, the need for Natural Language Processing (NLP) techniques becomes evident. This chapter aims to explore the foundational theoretical principles of NLP, providing a comprehensive overview. We will examine the subject from various perspectives, delving into each related aspect, ranging from a broad understanding to intricate details. The chapter will address the challenges associated with different languages and highlight the significance of NLP. Moreover, we will delve into sentiment analysis approaches and techniques, offering a clear definition of Sentiment Analysis and its diverse application fields, then we present an overview about Arabic language and it's categories and the difficulties of applying NLP techniques on the Arabic text. Additionally, we will categorize previous works based on the used techniques and approaches.

2. Natural Language Processing (NLP)

Natural Language Processing (NLP) is a subfield of Artificial intelligence (AI) related to computational linguistics, which is concerned with modeling language. NLP helps computers or machines to understand, manipulate, and explain human language as it is spoken and written. In this process, natural language is translated into structured data. Machines can do tasks such as speech recognition, sentiment detection or generating questions and responses. The natural language processing system is considered as pipeline because it contains several rational stages of data processing. Natural language is the language in which humans communicate with each other and which has evolved through use and repetition like Arabic and English. It is different from programming language that it can take the form of speech or writing. Nowadays, machines can understand, interact and do tasks based on the Natural language, like Amazon's Alexa, Apple's Siri.

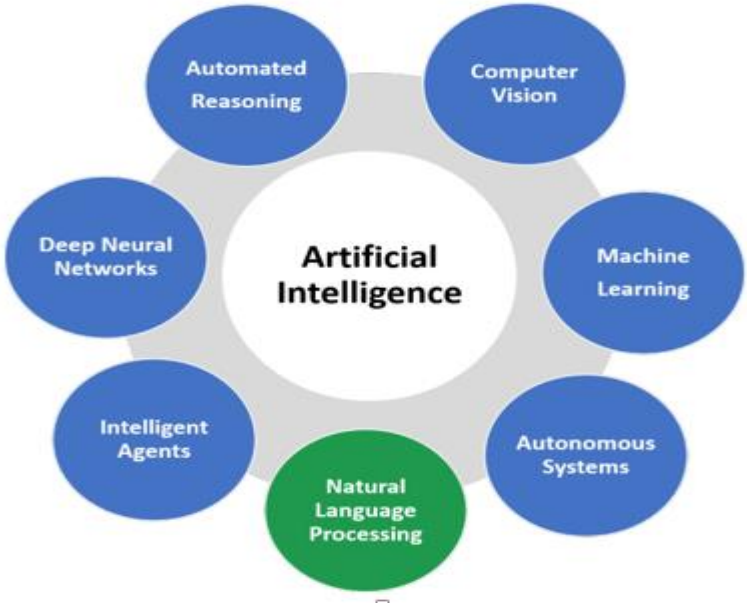

Figure 1.1- Position of NLP in AI

Natural language processing exists since 1940s, at that time NLP systems were implemented manually with code as a set of rules. Since 1990s most of the NLP researches are relied on machine learning. This latter provides automatic learning by using a huge amount of examples to build systems can deal with NLP, yet the biggest advantage of Machine learning for NLP is the accuracy [1].

2.1. NLP branches

We can define two aspects for NLP [2]:

2.1.1. Natural Language Understanding (NLU)

Natural Language Understanding is a branch of natural language processing that breaks down the elements of speech to assist computers comprehend and interpret human language. Machine learning models in natural language understanding (NLU) improve over time as they learn to detect syntax, context, linguistic patterns, unique meanings, sentiment and intent. Human-computer interaction is enabled through NLU.

2.1.2. Natural Language Generation (NLG)

Natural Language Generation is the use of artificial intelligence techniques to generate written or spoken narratives from a data set. This allows the chatbot to query data for repositories, such as integrated back-end systems and third-party databases, and use the results to generate a response.

2.2. NLP techniques

NLP is a rich field requiring the use of a number of different techniques in order to successfully process and understand human language. Below, we review and define a selection of the techniques commonly used in NLP technology.

2.2.1. Tokenization

Also called word segmentation, tokenization is one of the simplest and most important techniques involved in NLP. It is a crucial pre-processing step in which a long string of text is broken down into smaller units called tokens. Tokens include words, characters, and subwords. They are the building blocks of natural language processing, and most NLP models process raw text on the token level.

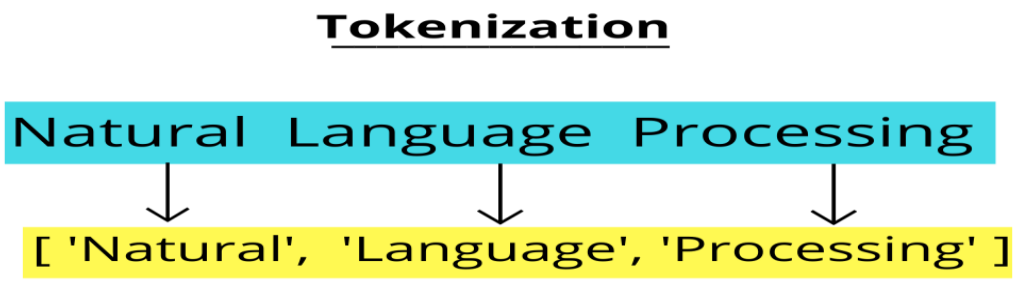

Figure 1.2- Tokenization [Web-1]

2.2.2. Stemming and Lemmatization

After tokenization, the next pre-processing step is either stemming or lemmatization. These techniques generate the root word from the different existing variations of a word.

For example, the root word "stick" can be written in many different variations, like:

- Stick
- Stuck
- Sticker
- Sticking
- Sticks
- Unstick

Stemming and lemmatization are two different ways to try to identify a root word.Stemming works by removing the end of a word. This NLP technique may or may not work depending on the word. For example, it would work on "sticks" but not "unstick" or "stuck".

Lemmatization is a more sophisticated technique that uses morphological analysis to find the base form of a word, also called a lemma.

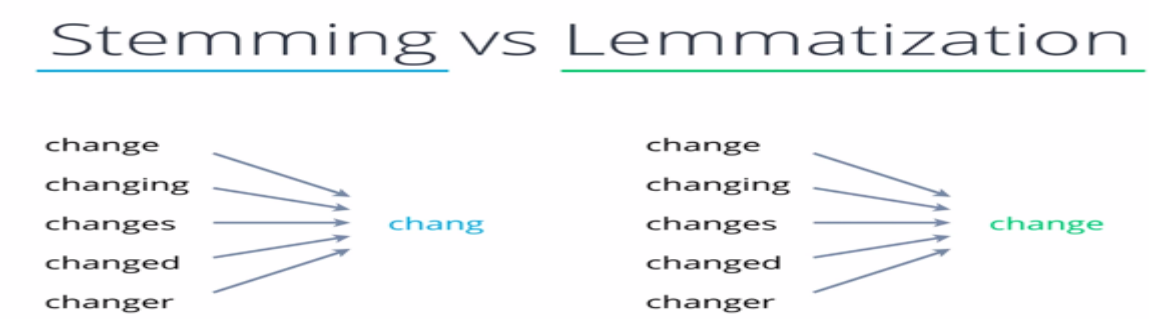

Figure 1.3- Stemming and Lemma using different forms of the word "change" [Web-2]

2.2.3. Morphological segmentation

Morphological segmentation is the process of splitting words into the morphemes that make them up. A morpheme is the smallest unit of language that carries meaning. Some words such as "table" and "lamp" only contain one morpheme. But other words can contain multiple morphemes. For example, the word "sunrise" contains two morphemes: sun and rise. Like stemming and lemmatization, morphological segmentation can help preprocess input text.

Figure 1.4- Morphological segmentation by breaking the word "unachievability" into its morphemes.

2.2.4. Stop words removal

Stop words removal is another pre-processing step of NLP that removes filler words to allow the AI to focus on words that hold meaning. This includes conjunctions such as "and" and "because," as well as prepositions such as "under" and "in".

By removing these unhelpful words, NLP systems are left with less data to process, allowing them to work more efficiently. It is not a necessary step of every NLP use case, but it can help with things such as text classification.

Table 1.1- Short phrases examples with the stop words removed [Web-3]

Sample text with Stop Words	Without Stop Words
GeeksforGeeks — A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal , Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

2.2.5. Parsing

Parsing is the process of figuring out the grammatical structure of a sentence, determining which words belong together as phrases and which are the subject or object of a verb. This NLP technique offers additional context about a text in order to help with processing and analyzing it accurately.

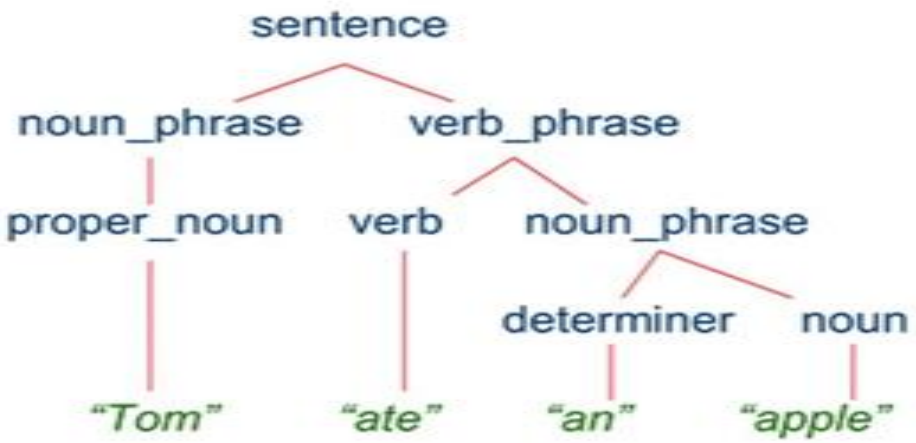

Figure 1.5- This is how parsing might work on a short sentence

2.3. NLP Applications

Humans pick up their natural language through the years with practice and repetition. This language can be difficult to decipher because it contains expressions and sentiments that go beyond literal meanings. That is why we often do not appreciate the complexity of our language

in the eyes of the machine and the difficulty of NLP application. As illustrated in Figure 1.6 NLP has numerous applications in diverse domains in both understanding or generating natural language, like sentiment analysis, emotion detection, plagiarism detection, behavioral detection, and fact checking.

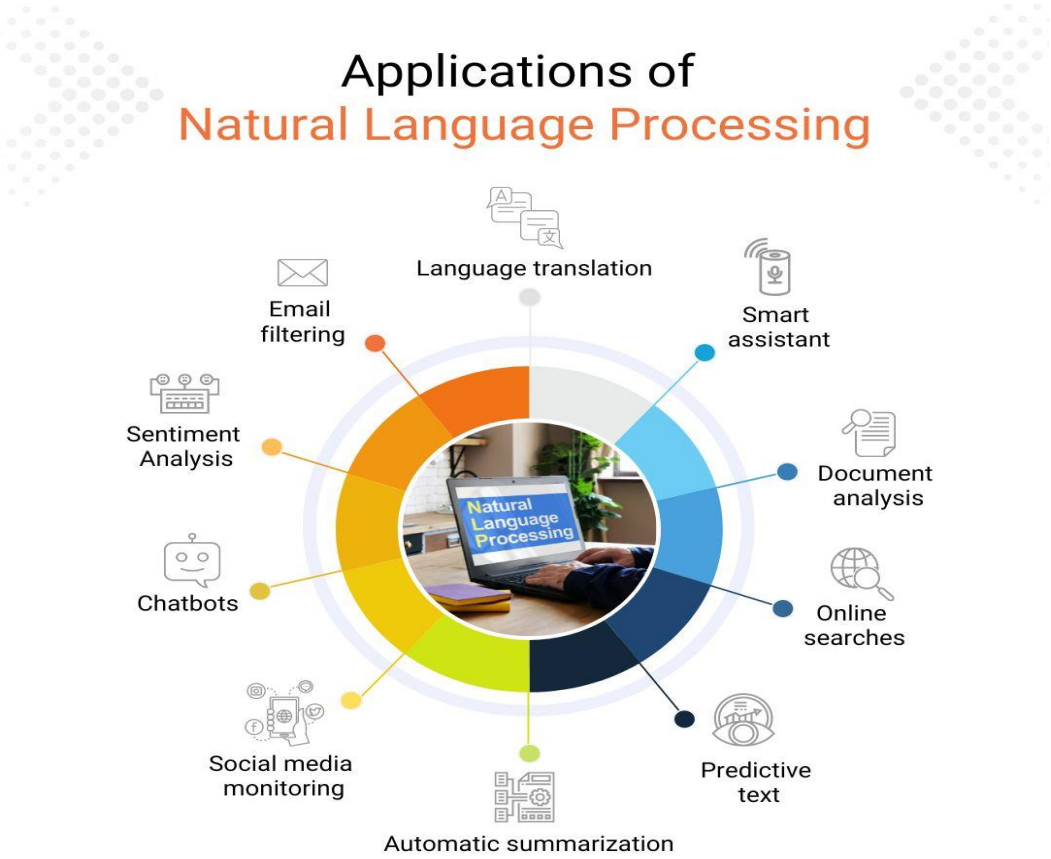

Figure 1.6- NLP applications

2.3.1. Email filtering

Email is a part of our everyday life. Whether it is related to work or studies or many other things, we find ourselves plunged into the pile of emails. We receive all kinds of emails from various sources; some are work-related or from our dream school or university, while others are spam or promotional emails. Here Natural Language Processing comes to work. It identifies and filters incoming emails into “important” or “spam” and places them into their respective designations.

2.3.2. Language translation

There are as many languages in this world as there are cultures, but not everyone understands all these languages. As our world is now a global village owing to the dawn of technology, we

need to communicate with other people who speak a language that might be foreign to us. Natural Language processing helps us by translating the language with all its sentiments.

2.3.3. Smart assistants

In today's world, every new day brings in a new smart device, making this world smarter and smarter by the day. And this advancement is not just limited to machines. We have advanced enough technology to have smart assistants, such as Siri, Alexa, and Cortana. We can talk to them like we talk to normal human beings, and they even respond to us in the same way.

All of this is possible because of Natural Language Processing. It helps the computer system understand our language by breaking it into parts of speech, root stem, and other linguistic features. It not only helps them understand the language but also in processing its meaning and sentiments and answering back in the same way humans do.

2.3.4. Document analysis

Another one of NLP's applications is document analysis. Companies, colleges, schools, and other such places are always filled to the brim with data, which needs to be sorted out properly, maintained, and searched for. All this could be done using NLP. It not only searches a keyword but also categorizes it according to the instructions and saves us from the long and hectic work of searching for a single person's information from a pile of files. It is not only limited to this but also helps its user to inform decision-making on claims and risk management.

2.3.5. Online searches

In this world full of challenges and puzzles, we must constantly find our way by getting the required information from available sources. One of the most extensive information sources is the internet. We type what we want to search and checkmate! We have got what we wanted. But have you ever thought about how you get these results even when you do not know the exact keywords you need to search for the needed information ? Well, the answer is obvious.

It is again Natural Language Processing. It helps search engines understand what is asked of them by comprehending the literal meaning of words and the intent behind writing that word, hence giving us the results, we want.

2.3.6. Predictive text

A similar application to online searches is predictive text. It is something we use whenever we type anything on our smartphones. Whenever we type a few letters on the screen, the keyboard gives us suggestions about what that word might be and when we have written a few

words, it starts suggesting what the next word could be. These predictive texts might be a little off in the beginning.

Still, as time passes, it gets trained according to our texts and starts to suggest the next word correctly even when we have not written a single letter of the next word. All this is done using NLP by making our smartphones intelligent enough to suggest words and learn from our texting habits.

2.3.7. Automatic summarization

With the increasing inventions and innovations, data has also increased. This increase in data has also expanded the scope of data processing. Still, manual data processing is time taking and is prone to error. NLP has a solution for that, too, it can not only summarize the meaning of information, but it can also understand the emotional meaning hidden in the information. Thus, making the summarization process quick and impeccable.

2.3.8. Sentiment analysis

The daily conversations, the posted content and comments, book, restaurant, and product reviews, hence almost all the conversations and texts are full of emotions. Understanding these emotions is as important as understanding the word-to-word meaning. We as humans can interpret emotional sentiments in writings and conversations, but with the help of natural language processing, computer systems can also understand the sentiments of a text along with its literal meaning.

2.3.9. Chatbots

With the increase in technology, everything has been digitalized, from studying to shopping, booking tickets, and customer service. Instead of waiting a long time to get some short and instant answers, the chatbot replies instantly and accurately. NLP gives these chatbots conversational capabilities, which help them respond appropriately to the customer's needs instead of just bare-bones replies.

Chatbots also help in places where human power is less or is not available round the clock. Chatbots operating on NLP also have emotional intelligence, which helps them understand the customer's emotional sentiments and respond to them effectively.

2.3.10. Social media monitoring

Nowadays, every other person has a social media account where they share their thoughts, likes, dislikes, experiences, etc., which tells a lot about the individuals. We do not only find

information about individuals but also about the products and services. The relevant companies can process this data to get information about their products and services to improve or amend them. NLP comes into play here. It enables the computer system to understand unstructured social media data, analyze it and produce the required results in a valuable form for companies.

3. Sentiment analysis (SA)

3.1. Definition

Sentiment analysis or Opinion Mining [3] is a field of study that aims to detect opinions or emotions expressed in text, along with their corresponding targets, whether expressed explicitly or implicitly. To delve into further detail, the essence of this domain can be understood by examining the problems it seeks to address:

- Given a text expressing an opinion about an object, the goal is to determine the polarity of the opinion. This can be achieved by either categorizing it into one of two opposing polarities (classification) or positioning it along a continuum between those two polarities (regression).
- Classifying a text as either subjective or objective.
- Identifying the topics present in a text and the opinions associated with them.
- Recognizing humor, irony, and sarcasm expressed within a text.
- Determining the author of a given text. [4]

3.2. Short history

Sentiment analysis as a term appeared for the first time in the article "Sentiment analysis: capturing favorability using natural language processing" [5] published in 2003 by the IBM researchers Nasukawa and Yi. The research field is also known as "opinion mining", term that appeared in the same year in the article "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews" [6] published by Dave et al. Other names include: opinion extraction, sentiment mining, subjectivity analysis [7]. The last one is justified by the observation that sentences expressing opinions are inherently subjective [8]. Given that most of the work so far focused on the written text, sentiment analysis is considered a subfield of the NLP. Although previous work existed, the field started to gain popularity and interest from the research world at the beginning of this century, due to several factors. One is the emergence of the social web, which led to an abundance of opinions saved in digital format that can be used as data by the researchers in this field. The other is the development and use of automatic

learning algorithms in NLP and information retrieval. Existing applications of sentiment analysis include: understanding consumer feedback, review summarization, brand analysis and reputation management, finding the emotional state of a nation, public opinion understanding, event monitoring, media studies, and financial predictions (there are studies stating the correlation between a nation's emotional state and stock price changes) [9].

3.3. Sentiment analysis levels

Sentiment analysis can occur at different levels: document level, sentence level or aspect/feature level.

3.3.1. Document level

In this process, sentiment is extracted from the entire review, and a whole opinion is classified based on the overall sentiment of the opinion holder. The goal is to classify a review as positive, negative, or neutral.

Example

"I bought an iPhone a few days ago. It is such a nice phone, although a little large. The touch screen is cool. The voice quality is clear too. I simply love it!" Is the review classification positive or negative? Document level classification works best when the document is written by a single person and expresses an opinion/sentiment on a single entity.

3.3.2. Sentence level

This process usually involves two steps:

- Subjectivity classification of a sentence into one of two classes: objective and subjective.
- Sentiment classification of subjective sentences into two classes: positive and negative.

An objective sentence presents some factual information, while a subjective sentence expresses personal feelings, views, emotions, or beliefs. Subjective sentence identification can be achieved through different methods such as Naïve Bayesian classification. However, just knowing that sentences have a positive or negative opinion is not sufficient. This is an intermediate step that helps filter out sentences with no opinions and helps determine to an extent if sentiments about entities and their aspects are positive or negative. A subjective sentence may contain multiple opinions and subjective and factual clauses.

3.3.3. Aspects/ Features level:

It performs a finer analysis than the other levels. It is based on the idea that an opinion consists of a feeling and a target, it identifies and extracts object features that have been commented on by the opinion holder and determines whether the opinion is positive, negative, or neutral. This level of analysis makes it possible to differentiate the aspects which are liked or not by the authors of the texts and thus makes it easier to determine possible treatment.

Sentiment analysis is extremely useful in social media monitoring because it provides an overview of the public's opinion on certain topics. The ability to extract insights from social web data is a practice that is widely adopted by companies around the world, therefore, the use of sentiment analysis is both broad and powerful.

3.4. Sentiment analysis applications

The importance of sentiment analysis is present in various domains, such as politics, medical field, emergencies, economy, security and sociology, so several applications have emerged in this context. In the following section we will mention some applications of sentiment analysis in various fields.

3.4.1. Online Commerce

One of the primary applications of sentiment analysis is in online commerce. Websites enable users to share their shopping experiences and provide feedback on product quality. These platforms offer product summaries and highlight various features by assigning ratings or scores. Customers can easily access opinions and recommendations about both the overall product and specific product attributes. Users are presented with graphical summaries that depict the overall product sentiment and its features. Prominent e-commerce platforms like amazon.com not only include reviews from editors but also incorporate feedback from customers, along with rating information. Another popular website, http://tripadvisor.in, offers reviews on hotels and travel destinations, with a massive database of 75 million worldwide opinions and reviews. Sentiment analysis plays a crucial role in assisting such websites by analysing this vast amount of feedback, helping them convert dissatisfied customers into enthusiastic promoters.

3.4.2. Voice of the Market (VOM)

The Voice of the Market concept revolves around understanding customer sentiments regarding competitors' products or services. Accurate and timely information obtained through the Voice of the Market is instrumental in gaining a competitive edge and driving new product development. Detecting such information at an early stage enables companies to directly target

key marketing campaigns. Sentiment analysis plays a crucial role in providing real-time customer opinions to corporations. This real-time information assists them in designing effective marketing strategies, enhancing product features, and predicting potential product failures

3.4.3. Voice of the Customer (VOC)

Voice of the Customer is concern about what individual customer is saying about products or services. It means analysing the reviews and feedback of the customers. VOC is a key element of Customer Experience Management. VOC helps in identifying new opportunities for product inventions. Extracting customer opinions also helps identify functional requirements of the products and some non-functional requirements like performance and cost.

3.4.4. Brand Reputation Management

Brand Reputation Management involves the proactive management of one's reputation in the market. Opinions expressed by customers or other parties can either harm or enhance a brand's reputation. Unlike customer-centric approaches, Brand Reputation Management (BRM) focuses on the product and company itself. With the prevalence of one-to-many conversations occurring online at a rapid pace, organizations now have the opportunity to effectively manage and strengthen their brand reputation. Brand perception is no longer solely influenced by advertising, public relations, and corporate messaging. Rather, brands are now shaped by the collective conversations surrounding them. Sentiment analysis plays a crucial role in determining how a company's brand, product, or service is perceived within the online community.

3.5. Sentiment analysis approaches

Sentiment analysis field has been well studied by researchers in the past few years. Many different methods and techniques have been developed and tested through different tasks and at different levels. However, a lot of work is yet to be done. Sentiment analysis is the contrary to simple text classification due to the many challenges of the field, here three types of techniques had been used to classify opinions, which are: Lexicon-Based approach, machine learning based approach, and hybrid approach.

3.5.1. Lexicon-Based approach

The lexicon-based approach depends on finding the opinion word lexicon which is used to analyse the text, It identifies the polarity of a text using two sets of words divided into, positive

represent the desired expression and negative represent the undesired expression, it require the sentiment lexicon to generate it either manually or semi-automatically.

The model counts in the text the number of positive words and the number of negative words, the sum gives an overall evaluation of the feeling of the text. The input text will be converted to tokens by the Tokenizer of NLP system, every token will get compared with the lexicon in the dictionary or in the corpus. If there is a positive mark, the result will be added to the total pool of score for the input text, and then the total score of the text is incremented. Else, the score is decremented or the word is tagged as negative, the text is possibly neutral if the numbers are equal.

This technique is governed by the use of two methods: the dictionary-based approach [10], and corpus-based approach [11]. Both approaches could be done by using statistical or semantic methods. The dictionary-based approach depends on finding opinion root or seed words, and then searches the dictionary of their synonyms and antonyms. The corpus-based approach starts with a list of seed opinion words, and then finds other opinion words in a large corpus to find opinion words with context specific orientations.

3.5.2. Machine learning approach

Sentiment classification and categorizing text into positive, negative or neutral categories require more practical techniques, hence the use of the machine learning technique with their fully automatic implementation and an ability to handle large collections of data. Machine learning techniques are most useful techniques for sentiment analysis, because the steps of training a dataset by learning the documents then testing it to validate the performance, are very powerful and accurate. There are a number of machine learning algorithms used to classify texts.

Machine Learning-Based Sentiment classification methods can be categorized into: supervised and unsupervised. In sentiment analysis the most used is supervised learning, it generally comprises of four steps: Data collection, Pre-processing, training data, Classification and writing results.

It begins with the collection of training data as tagged corpus, then the classifier will be trained on this data and present a series of feature vectors from it. Therefore, a model will be created based on the training dataset which will be used over the new text for classification

purposes. Results will be written based on the type of representation selected, the performance tuning and the execution precision are done before the release of the algorithm.

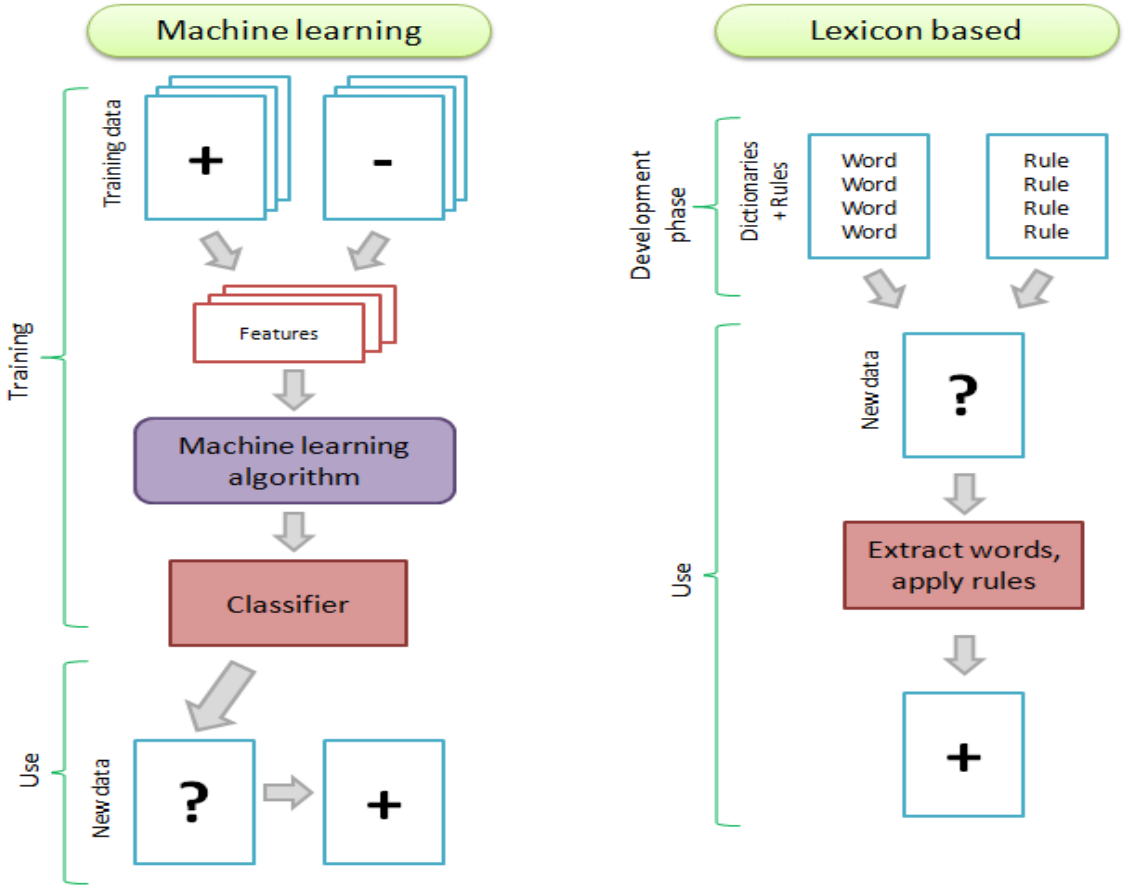

Figure 1.7- A comparison of automated and lexicon-based sentiment analysis methods [12]

3.5.3. Hybrid approach

Hybrid approach combines the strengths of the two previous approaches, it could collectively expose the accuracy of a machine learning approach and the speed of lexical approach. This approach takes into account all the processing linguistics of the lexical approach before starting the learning process as in statistical approaches. The hybrid approach gives the high accuracy from the machine learning and the stability from the lexicon-based approach.

The lexicon based approach tools and techniques use dictionaries and lexicons as the major source of search for sentiment classification. These lexicons have seeded semantic orientations that are later compared with the input dataset for classification. Machine learning based approaches instead follow the learning algorithms to create the training dataset. Then on the basis of this trained dataset the inputs are compared and classified as either positive, negative or any other sentiment. Here we present.

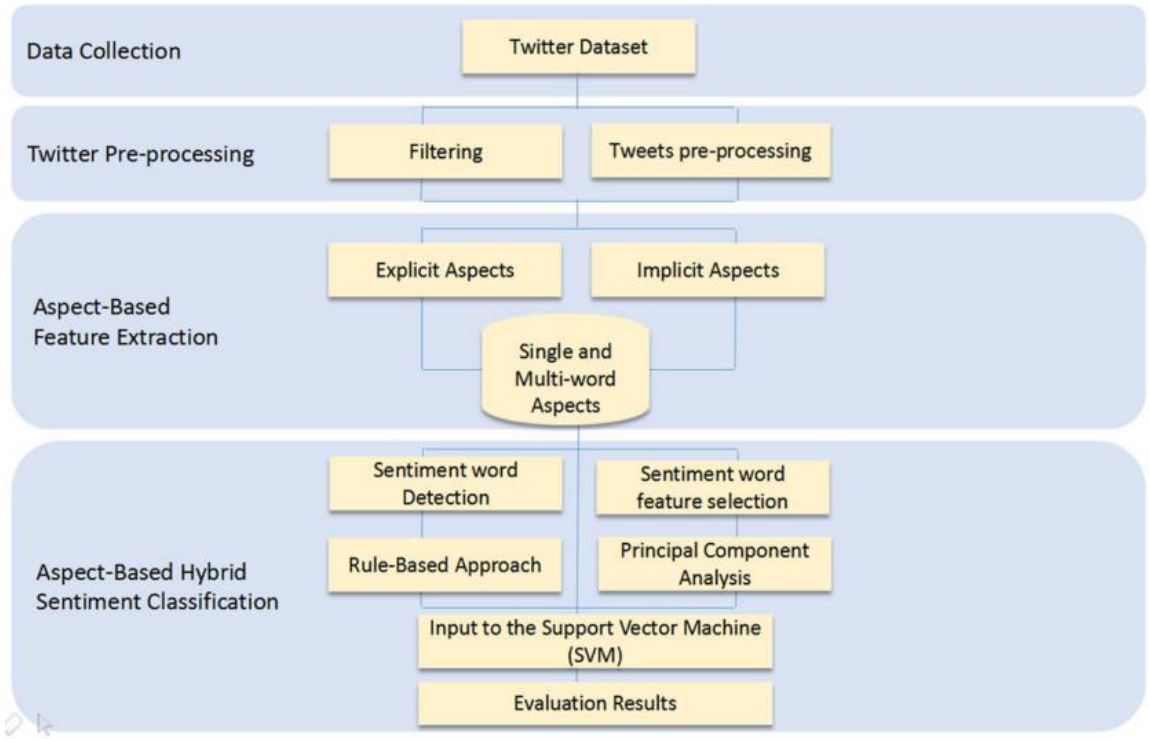

Figure 1.8- An example of hybrid sentiment classification [13]

3.6. Sentiment analysis challenges

Sentiment analysis classifies text as positive, negative or else objective, so it can be thought as text classification task. Text classification has many classes as there are many topics but sentiment analysis has only three classes. However, there are many factors that make sentiment analysis difficult compared to traditional text classification. The following are some of the factors.

3.6.1. Coreference resolution

Coreference Resolution is the problem of identifying what a pronoun, or a noun phrase refers to. For example, "We watched the movie and went to dinner; it was awful." What does "It" refer to? Coreference resolution may be useful for the topic/aspect based sentiment analysis. Coreference resolution may improve the accuracy of opinion mining.

3.6.2. Temporal Relations

The time of reviews may be important for sentiment analysis. The reviewer may think that Windows Vista is good in 2008, but he may have negative opinion in 2009 because of new Windows 7.So assessing this kind of opinions that are changed with time may improve the performance of the sentiment analysis system. This helps us to observe if a certain product gets improved with time, or people change their opinion about a product.

3.6.3. Sarcastic sentences

Text may have Sarcastic and ironic sentences. For example, “What a great car, it stopped working in the second day.” In such case, positive words can have negative sense of meaning. Sarcastic or ironic sentences can be hard to identify which can lead to erroneous opinion mining.

3.6.4. Domain Considerations

The accuracy of sentiment classification can be influenced by the domain of the items to which it is applied. The reason is that the there are many words whose meaning changes from domain to domain. For example “Go read the book.” This sentence has positive sentiment in book domain while it indicates negative sentiment for movie domain.

3.6.5. Grouping synonyms

Many times text contains different words having same meaning. So such word should be identified and group together for accurate classification. It is a difficult task to identify these words, as people often use different words to describe the same feature. For example, “voice” and “sound” both refer to the same feature in phone review.

3.6.6. Thwarted Expectations

Some texts contain sentences that start with a context, and end in a different context. For example, “The cast was not good, actors performed poorly, but I liked it.” In above review the last sentence makes the whole review positive. If term frequency considered the above statements would classify as negative due to more negative words in review.

In conclusion, sentiment analysis encounters various challenges that need to be addressed for accurate and effective analysis.Additionally, the challenges faced in sentiment analysis are further amplified when applied to Arabic language text.Arabic sentiment analysis presents unique difficulties due to the intricacies of the language itself. Factors such as morphology, dialectal variations, and the extensive use of figurative expressions make it more complex to accurately analyse sentiment in Arabic. Moreover, the scarcity of labeled datasets and the need for a deeper understanding of cultural nuances and contextual influences pose additional obstacles.

4. Arabic NLP (ANLP)

Since sentiment analysis heavily depends on the morphology of the language being analysed, here we a concise overview of Arabic and elucidate its linguistic characteristics that pose significant challenges for researchers in the field of sentiment analysis.

Arabic holds the distinction of being one of the six official languages recognized by the United Nations. With its status as the official language in 26 countries, it is spoken by a vast population of over 422 million individuals across the Arab world. Online, Arabic stands at an impressive fourth position among the most commonly used languages and has experienced remarkable growth in recent years, with a staggering 6091.9% increase in the number of Internet users [14].

The Arabic language encompasses three primary variations: Classical Arabic, which serves as the language of the Quran, the sacred text of Islam; Modern Standard Arabic (MSA), widely employed in written form and formal speeches; and dialectal Arabic, which refers to the various spoken forms used in everyday communication. The dialects exhibit regional variations both between different Arab countries and within specific regions of the same country [14].

4.1. Arabic Orthography

In contrast to Latin languages, Arabic is written from right to left and stands out due to the absence of capitalization or lowercase letters. Its alphabet comprises 28 letters, including 25 consonants and only 3 vowels. Alongside these vocal elements, Arabic writing employs diacritical marks, such as short vowels. These marks are positioned above or below the letters to ensure accurate pronunciation and clarify the meaning of words. Most Modern Standard Arabic (MSA) texts are written without short vowels, as proficient speakers do not rely on diacritical marks to comprehend the text. However, diacritical marks are frequently used in children's books and educational materials for learning Arabic. The absence of diacritical marks in most texts presents a challenge of lexical ambiguity, as it can give rise to multiple interpretations. For instance, (شعر) pronounced as "sha'r" denotes "hair," whereas (شعر) pronounced as "shi'r" means "poetry."

4.2. Arabic Grammar

Arabic grammar distinguishes between two types of sentences: verbal and nominal. Verbal sentences typically begin with a verb, while nominal sentences begin with a noun or a pronoun. These sentences can exhibit syntactic flexibility, allowing for a wide range of syntactic variations and possibilities. They may also include multi-word expressions, such as "فقر الدم"

(Anemia) in medical terminology, which consists of two words with the literal meanings of "فقر" (poor) and "دم" (blood). Additionally, these sentences may contain arguments, adjectives, and pronominal anaphora, which introduce ambiguity regarding the presence of the third-person pronoun, known as "ضمير الغائب" in Arabic. Identifying the correct antecedent referred to by the third-person pronoun (her/hers/it/its, him/his/it/its, them/their) can be challenging, especially when it is hidden, leading to grammatical errors in automated Natural Language Processing (NLP) systems. These complexities make Arabic syntax intricate and recognized as a highly challenging problem in NLP.

4.3. Semantic Ambiguity

Arabic exhibits constituent boundary ambiguity, as seen in the phrase "مديرة المدرسة الجديدة" which can be interpreted as "The new school directress" or "The new directress of school" depending on the boundary of the adjective phrase within this noun construct. Furthermore, Arabic's semantic ambiguity is even more intricate and indefinite, where sentences and phrases can be interpreted in different ways. For example, in the sentence " يكره علي أحمد اكثر من ابراهيم " (Ali hates Ahmed more than Ibrahim.)), it could mean that Ali hates Ahmed more than Ali hates Ibrahim, or it could imply that both Ali and Ibrahim hate Ahmed, but Ali hates Ahmed more than Ibrahim hates Ahmed.

Finally, in addition to all these challenges and problems in ANLP, many other difficulties will increase when it comes to dialect, especially that in social media the dialect is the most used. Every specific dialect makes changes and variations whether in lexical, phonology and morphology. This will make the sentiment analysis in social media more confrontational and challenging.

5. Sentiment analysis in Arabic

The unique structure of Arabic words is one of the main challenges researchers face when dealing with Arabic sentiment analysis. The following section explores some of the key challenges encountered in the efforts to establish an accurate system for Arabic.

5.1. Morphological analysis

Morphological analysis is a crucial phase in Sentiment Analysis. Its main objective is to break down words into morphemes and associate each morpheme with morphological information such as root, stem, Part Of Speech (POS), and affix. As we have seen in the previous section, Arabic is a morphologically complex language. This complexity calls for the development of

appropriate systems capable of handling tokenization, spelling verification, stemming, lemmatization, pattern matching, and part-of-speech tagging. Nowadays, numerous morphological analyzers for Arabic have already been developed, some of which are freely available while others are commercially oriented. Among those mentioned in the literature are Arabic Morphological Analysis and Generation.

5.2. Arabic dialect

For communication purposes, Arabic speakers commonly use colloquial Arabic rather than Modern Standard Arabic (MSA). There are approximately 30 major Arabic dialects that differ from MSA and from one another in terms of phonology, morphology, and lexicon [14]. Furthermore, Arabic dialects lack a standardised orthography and language academies. Consequently, using tools and resources designed for MSA to process Arabic dialects leads to significantly lower performance. Recently, researchers have started developing analyzers for specific dialects, such as CALIMA [15] for the Egyptian dialect. However, these analyzers still have low accuracy and are only designed for specific dialects. Bridging this gap in Arabic processing will enhance information retrieval efficiency, particularly for social media data.

5.3. Algerian Dialect

Nowadays, the Algerian Dialect (ALGD) is more often used in interviews, telephone conversations and public services. Moreover, ALGD is becoming very present in blogs, forums and online user comments. Therefore, it is important to consider this dialect in the context of NLP.

In comparison to MSA, Algerian dialect is considered less normalized and standardized. It has an Arabic-inspired vocabulary, but the initial words have been phonologically and morphologically changed [16]. In addition to the words from MSA, Algerian dialect is characterized by the presence of words borrowed from Tamazight, French, English, Italian, Turkish and Spanish. ALGD is also divided according to regions, for example into: East, West, South, Kabyl, Chaoui, Mzabi, and Tergui. Other new languages are also used due to culture fans for instance, Japanese, Korean and others. The following are some of the key characteristics of Algerian dialect in the web:

- **The use of numbers instead of letters or words**: with the advent of cell phones and the social network, people began to use more and more abbreviations. Numbers were

used to replace letters and syllables because they resembled certain letters and syllables. The Table 1.2 shown some examples.

Table 1.2- Use of numbers instead of letters or words example

Letter and its alternative number	**Example**
"ح" replaced with "7"	ta7t = تحت , that mean "under"
"ق" replaced with "9"	Fou9 = فوق , that mean "over"
"ع" replaced with "3"	3ami = عمي , that mean "uncul"
"خ" replaced with"5"	5abi = خبي ,that mean "hide"

- **Code switching:** North Africans alternate between two or more languages, or language varieties, in the context of a single conversation. This is illustrated in the following example: "Merci خويا". The user has used an Arabic expression "خويا" and a French expression "Merci" which means "Thank you Brother". However, the Algerian dialect is also formed by transformed words from the languages, which inspired Algerian through the ages. Take the word "نتا" which is inspired from the Arabic word "انت" meaning "you", where the first letter was changed from the beginning of word to the last. This phenomenon is known as "Intraword switching" in linguistics [17], where a switch could occur in one or more places in the same word.
- **Encoding a language in another language's letters:** either Arabic expressions encoded in Roman letters, known as "arabizi" or the reverse, known as "Romanization" As an example of arabizi, consider "9alb" which is written in Arabic as "قلب" and means "heart", and "باي" which is written in Arabic and refers to the English word "bye".

Above all, there is the potential for multiple languages to coexist in a single sentence. The Algerian dialect is exceedingly challenging to understand and very complex to process automatically due to the wide variety of writing styles, potential writing errors, and new words, frequently appearing. These linguistic diversities call for special attention, which is why the spoken and written dialects are very rich and varied languages

6. Related works

The existing related works on sentiment analysis can be classified from different points of views: used technique, view of the text (language), level of detail of text analysis, source of data, rating level and many others. In this section we present classification based on the previous

approaches which we identified (lexicon approach, machine learning approach, hybrid approach), social media as source of data and English also Arabic (MSA and DA) as text language.

Table 1.3- Related works

Applied Approach	Article	Language	Main Tools	Accuracy
Lexicon-based approach	[18]	English and German	LIWC	**
	[19]	Arabic (MSA)	Twitter's APIs	86.89%
	[20]	Arabic (MSA and Saudi dialects)	**	**
Machine Learning approach	[21]	Arabic (MSA)	Twitter's APIs, Rapidminer, TF / TF-IDF, n-gram (SVM)	73.15%
	[22]	Arabic (MSA and Moroccan dialects)	TF / TF-IDF, n-gram (SVM)	~78%
	[23]	Algerian dialect	CNN, SVM	**
Hybrid Approach	[24]	English	POS tagging, SentiWordNet, n-gram, KNN	86%
	[25]	Arabic (MSA and Saudi dialects)	Twitter's API SentiWordNet, n-gram, (SVM)	84.01%

7. Conclusion

Sentiment analysis is a process during which the polarity is positive, negative or neutral of a given text, it is usually applied on social media platforms, specifically on the user's posts, comment, tweets or even messages. In this chapter, we introduced a definition of sentiment

analysis. Also with explaining and clarifying the wide application fields of sentiment analysis and its effect on our daily lives, we also defined the different approaches of sentiment analysis.

While sentiment analysis is a process applied in texts written by humans, Natural Language processing (NLP) was an important title in our chapter, where we explained how the NLP system works, with its different techniques. Additionally we brought up some of the wide challenges faced by NLP in particularly Arabic languages and its different Dialectes. Finally, we classified some related works according to the used approach.

In the next chapter, we will difine machine learning and deep learning. Moreover, we will present the most used techniques and architectures in deep learning like RNNs and the Tronsformers

Chapter 2 : Deep Learning

1. Introduction

In recent years, Deep Learning (DL), which is a subset of Machine Learning, has revolutionized the field of Natural Language Processing (NLP) by improving the performance of various tasks based on NLP such as: machine translation, sentiment analysis , and answering questions.

This chapter provides the background knowledge required for this work. First, we start with a brief definition of machine learning and its common algorithmes. Second, we provides a basic background knowledge of deep learning and artificial neural networks, then we will present the common activation functions and loss functions. After that, we will present two types of specialized artificial neural networks, which are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Finally, we will describe on the Transformer architecture in detail, and based on this we present the BERT model.

2. Machine Learning (ML)

Machine Learning is a subfield of Artificial Intelligence. In order to solve many real world problems, the machine learning is one of the most widely used things of our time. It allows the computer to perform complex operations and predictions based only on data and without any human intervention or explicit coding. ML algorithms are therefore quicker and more potent than other techniques that call for hand-coding rules. The ML helps to develop algorithms or programs useful for understanding, managing, solving, and analyzing big and complex data problems, additionally problems that cannot be fixed manually. Machine learning is gaining elevation due to the increasing volume and variety of data, the accessibility and affordability of computing power, and the availability of high-speed internet.

2.1. Types of Machine learning

There are several methods for performing machine learning. They can be classified into five main categories according to their need for the intervention of the human actor. This leads to supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, and transfer learning which we will explain below. Self-learning and anomaly detection also considered types of learning with certain particularities.

2.1.1. Supervised learning

Supervised learning is concerned with predicting a target value given input observations. In machine learning, we call the model inputs "features". The target values that supervised models are trained to predict are also often called labels [26] [27].

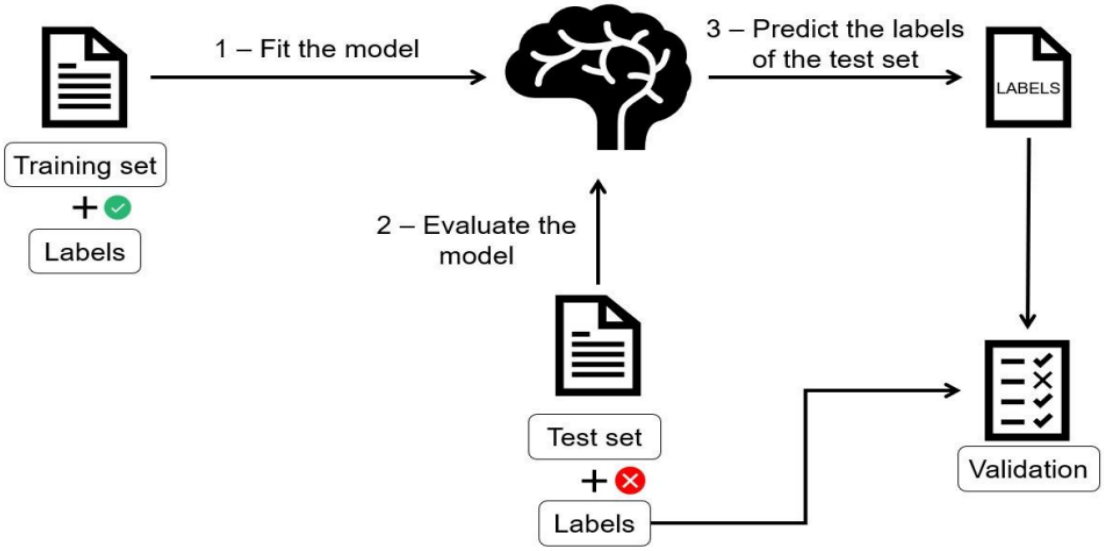

Figure 2.1- Supervised learning [28]

2.1.2. Unsupervised learning

This type of learning consists of feeding an algorithm with unlabeled data, its task will be to find the similarities between these data. This method is useful because unlabeled raw data is more abundant than structured data. This technique is used to discover hidden patterns in a data set or to automatically determine the characteristics of inputs. For example, it is particularly useful to use it to automatically label data.

2.1.3. Semi-supervised learning

Semi-supervised learning uses the two methods mentioned above, that is the inputs contain labeled and unlabeled data. This type of learning can improve the quality of an algorithm in certain cases but can also deteriorate it, it depends on the sensitivity of the algorithm in relation to the data.

2.1.4. Reinforcement learning

Reinforcement Learning is a type of Machine Learning, which allows machines to automatically determine the ideal behavior within a specific context, in order to maximize its performance. The reinforcement learning method aims at using observations gathered from the interaction with the environment to take actions that would maximize the reward or minimize the risk. To produce intelligent programs (also called agents) [Web-4].

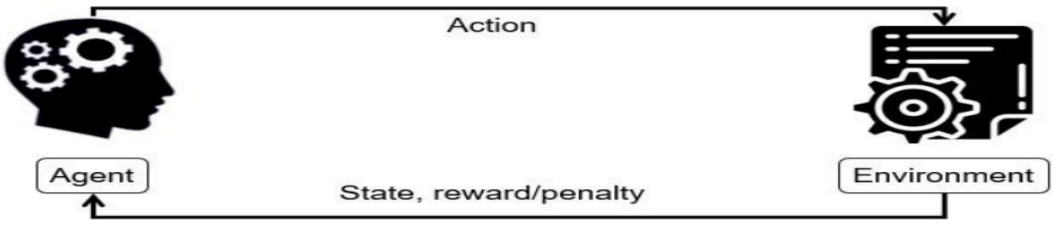

Figure 2.2- Reinforced learning [28]

2.1.5. Transfer Learning (TL)

Transfer learning is a machine learning technique where knowledge gained during training a set of problems (called source domain) with large dataset , can be used to solve other similar problems (target domain) with small dataset . In other term, TL can employ knowledge such as weights and features of the pre-trained model to train a new model, as well as undertake issues in the novel task that has a smaller amount of data. TL with deep learning models is more rapid, has improved accuracy, and needs less training data. The TL concept is to utilize a trained network on different tasks for different source data then adjust it for the target task (see Figure 2.3).

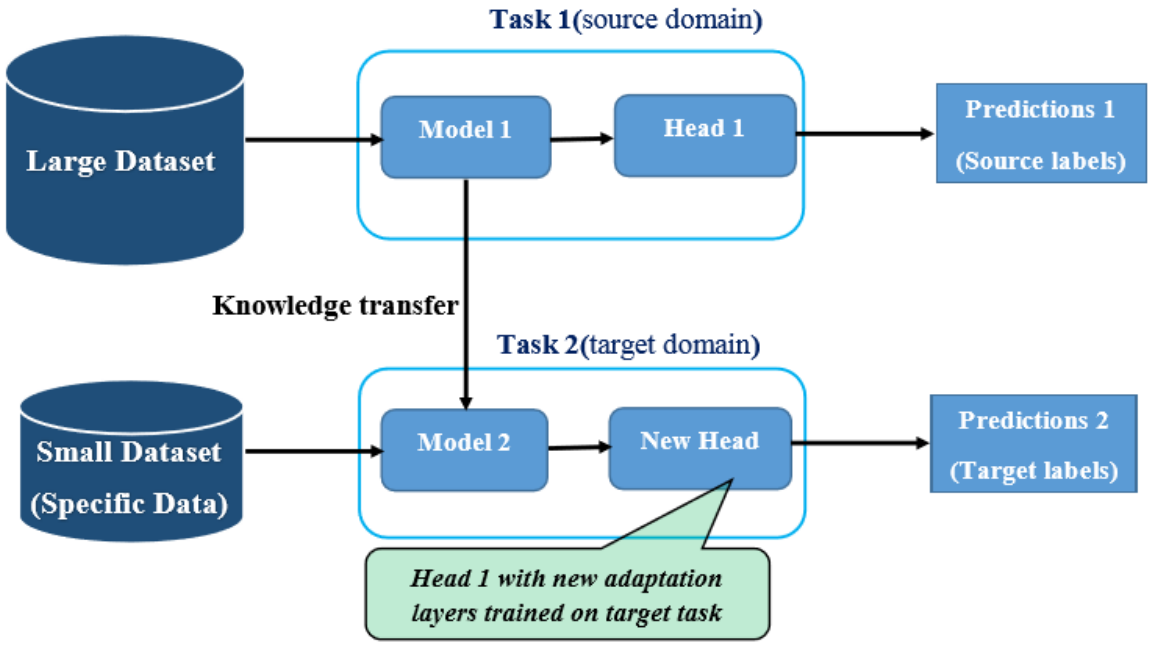

Figure 2.3- The concept of the transfer learning

3. Deep Learning (DL)

Deep learning (DL) is a subset of machine learning methodologies and techniques that use multiple hidden layers of the artificial neural networks (ANNs). It is the adaptation of neural networks that mimics the structure of the human brain [29]. The strength of DL lies in the fact that the machine can extract features and learn on its own, independent of the intervention of an expert. The deep learning methodology applies nonlinear transformations and model abstractions of high level in large databases [29].

4. Neural Networks (NNs)

Practically all DL algorithms are neural networks also called ANNs [30]. ANNs are information processing models that simulate the functioning of a biological nervous system. This is similar to how the brain handles information at the functioning level. All neural networks are made up of interconnected neurons that are organized in layers [30].

The detail of the neuron and the description of its operation will be explained in the next part

4.1. Biological neuron

The neuron, which is thought to be the fundamental building block of the central nervous system, is composed of a cell body called the soma that branches to produce what are referred to as dendrites. The dendrites carry the information through the cell body. The information will be transmitted to the other neurons through an extension called an axon. A synapse, which is an empty space between an axon and a dendrite, through which transmission takes place through chemicals [31].

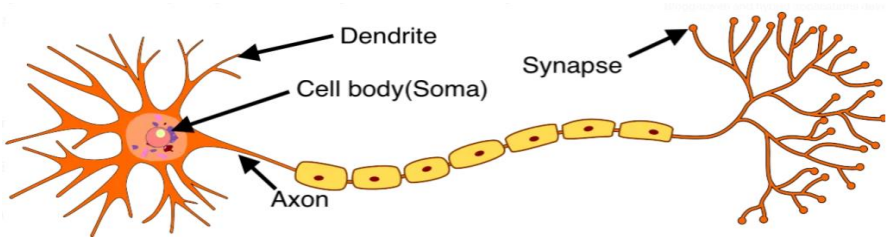

Figure 2.4- Biological neuron

The brain is made up of about one trillion nerve cells called neurons [32]. The output of each neuron is linked to thousands of other neurons. Each neuron performs local processing. It collects the signals from the dendrites and adds them up. If the resulting amplitude exceeds a certain internal threshold, a signal is sent through to other neurons.

4.2. Artificial neuron

An artificial neuron is a calculation model whose construction was influenced by how actual neurons work. This formal neuron can be thought of as an operator that receives a variety of inputs from the environment or from other neurons. Each of these inputs is weighted by a weight called synaptic weight, and it only produces an output when the sum of these inputs is greater than a specific internal threshold.

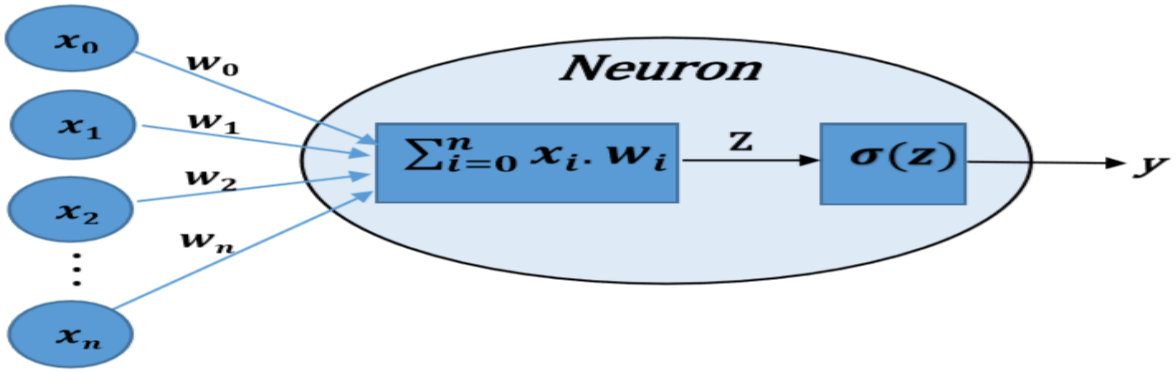

Figure 2.5- Artificial neuron

Neuron has multiple inputs (x_1,x_2,……,x_n), associated with their weights (w_1,w_2,……,w_n), in addition to a bias input x_0 =1 and its weight w_0,and only one output y is calculated via application of an activation function, for example,sigmoid (σ) on z which is the sum of multiplication of the inputs with their weights (see Figure 2.5).

Artificial neural networks consist of layers, which can be as follows:

- **Input Layer:** The first layer is the input layer; the input layer consists of a vector representing the data in its vectorized form.
- **Hidden layer:** The hidden layer are intermediate layer between the input and the output layer.
- **Output layer:** It produces the neural network's results based on given inputs.

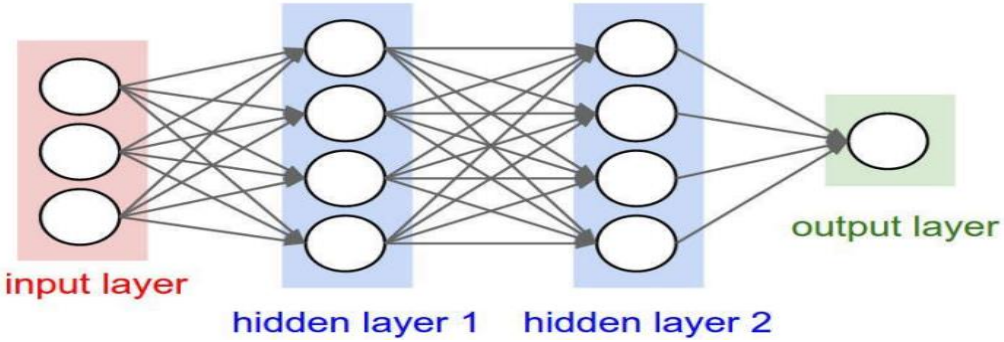

Figure 2.6- Neural networks layers

4.3. Activation Functions

For neural networks to function properly, Activation Functions are essential, especially when there are multiple layers. The weights of each stratum are identical. This is accomplished by calculating the gradient of the global loss function concerning each weight vector.

For the gradient to be computed effectively, it must traverse each layer. The gradient's value is determined by how we convert the layer's input. To keep training with the proper gradient, it would also be preferable to provide non saturating activation functions [33]. These are a few instances of frequently used activation functions (See Figure 2.7).

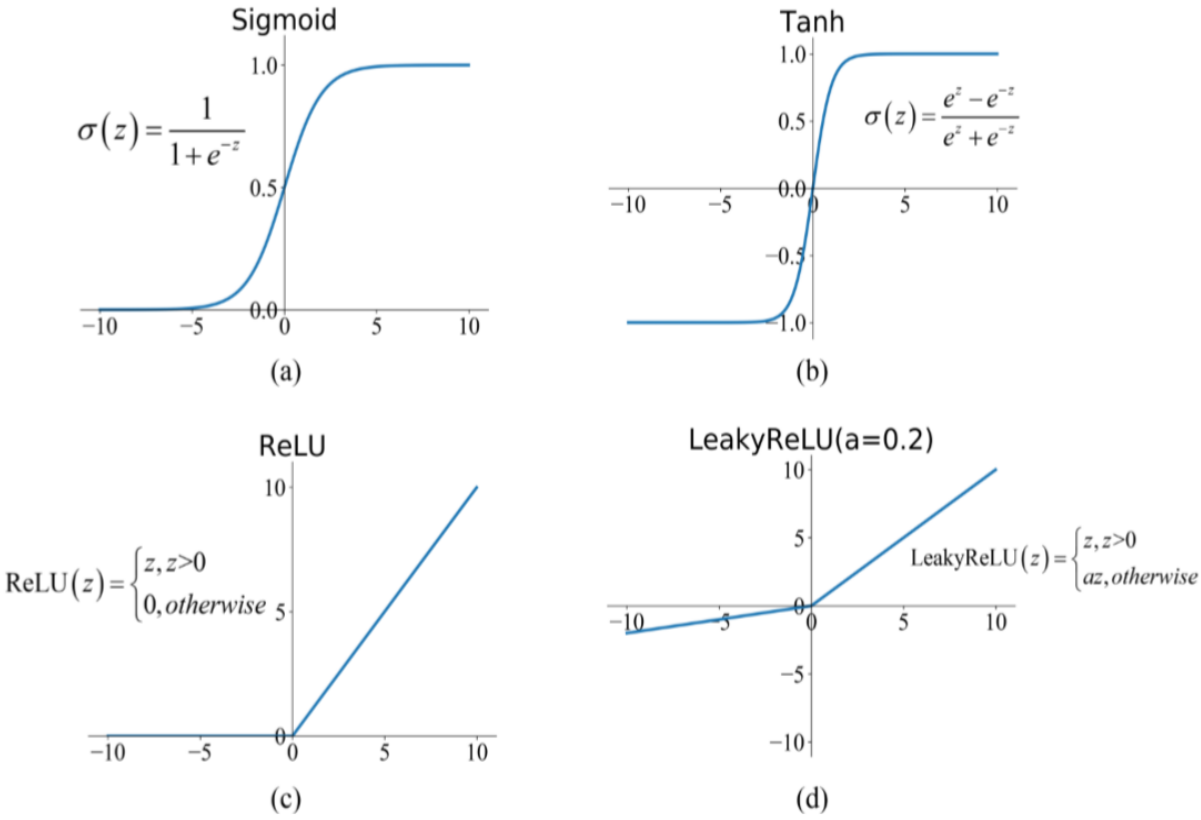

Figure 2.7- Commonly used activation functions [34]

A. Logistics function (Sigmoid): This is one of the most frequently utilized functions. It is commonly referred to as the logistic function or the logistics sigmoid, and it is limited between 0 and 1, and it may be understood stochastically as the probability that the neuron activates. It is defined as:

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

B. Hyperbolic Tangent (TanH): The TanH function is a trigonometric function whose waveform gives it the privilege of being selected among the activation functions. It is

the same as the logistic sigmoid function, but better the value is between 1 and -1. After differentiation, the value of this function becomes less than 1. It is defined as:

$$\sigma(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \quad (2)$$

- The relation between TanH function and Sigmoid function is :

$$TanH(z) = 2\,sigm(2z) - 2 \quad (3)$$

C. Rectified Linear Unit (ReLU): The ReLU function is likely the one that is most similar to its biological counterpart. Recently, many jobs (particularly those involving computer visions) have started to favor this function [35]. As in the formula bellow, this function returns 0 if the entry z is less than 0 and returns z it self ,if it is greater than 0 as shown on the Figure 2.7. It is defined as:

$$ReLU(z) = max(0, z) \quad (4)$$

D. Leaky-ReLU: Leaky Rectified Linear Unit is a type of activation function based on a ReLU that tries to solve the problem known as Dying ReLU.

A traditional rectified linear unit ReLU(z) returns 0 when $z \leq 0$. The Dying ReLU problem refers to when the unit gets stuck this way–always returning 0 for any input, Leaky ReLU aims to fix this by returning a small, negative, non-zero value instead of 0, as such :

$$LeakyReLU_a(z) = max(az, z) \quad (5)$$

E. GAUSSIAN ERROR LINEAR UNIT (GELU): This function solves most of the previous activations function issues and more importantly avoids the vanishing gradients problem. It provides a well defined gradient in the neg ative area and prevents neurons from dying; besides it performs best in transformers models [Web-5]. The GELU is formula approximated by:

$$GELU(z) = 0.5z(1 + \tanh[\sqrt{2/\pi}(z + 0.044715z^3]) \quad (6)$$

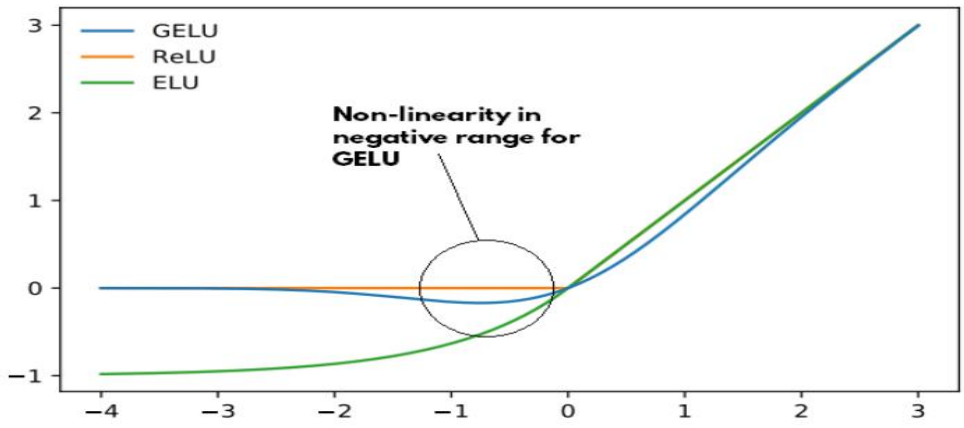

Figure 2.8- GELU [Web-5]

F. Softmax: Softmax is a special activation function for classification networks with multi-class problems.The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1. It is defined as :

$$Softmax(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{7}$$

4.4. The cost (loss) functions

In order to update the weights tying the neurons together, neural networks use a backpropagation technique based on the cost function. A cost function or the error function is how the global network did, a single value generally measured via the average difference between the output and the expected output of a training sample. We can conclude that cost function depends on the network weights, the biases of the network, and one training sample with the expected value of that training sample. Examples of commonly used cost functions :

- **Mean Squared Error (MSE):** Also known as the Quadratic cost function formula or maximum likelihood, it is the default choice for regression problems. Equation (8) explains how to calculate it.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{8}$$

Where N is the total number of inputs in the training sample, *y* the observed target

value, and $\hat{y}$ the predicted value.

- **Cross-entropy cost function (CE):** Also known as negative log likelihood loss is a commonly used loss function in machine learning for classification problems. The function measures the difference between the predicted probability distribution and the true distribution of the target variables. The formula to compute cross-entropy is described in the Equation (9).

$$CE = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \quad (9)$$

Where C is the number of classes given in the dataset, y the observed target value, and $\hat{y}$ the predicted value.

5. Deep Learning Algorithms

5.1. Convolutional Neural Networks (CNN)

A convolutional neural network is a special type of feed-forward neural network introduced in 1989 [36], and originally used in areas such as computer vision, and recommender systems. A CNN architecture typically consists of multiple alternate convolution and pooling layers to extract features from the image, followed by one or more fully connected layers at the end, allowing image classification from the extracted features (see Figure 2.9).

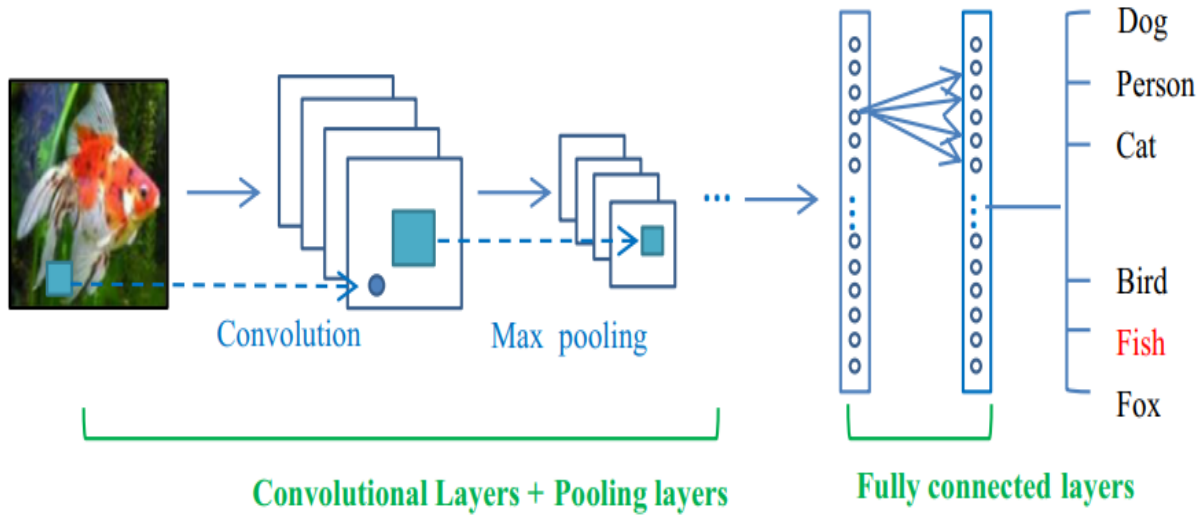

Figure 2.9- Typical CNN architecture [37]

5.1.1. Convolution Neural Networks Layers

A). Convolutional layer (CONV):

The convolution is a mathematical operation that consists of applying a filter (kernel) to an n- dimensional array (image). The filter is also an array of numbers that are called weights or parameters.

As the filter is sliding, or convolving, around the input image, it is multiplying the values in the filter with the original pixel values of the image. In other terms, it computes element-wise multiplication. The results of these multiplications are all summed up. Thus, the sliding of the filter along all the image gives us one filtered image (called **activation/feature map**).We should note that in this layer we apply many filters to the initial image and each filter constitutes a set of weights that we are learning during the training process. A convolution layer may be followed by an activation function layer (ReLu function).

To better understand how the convolution layer works, below is a simple example. We apply a filter to a 7×7 pixel image.

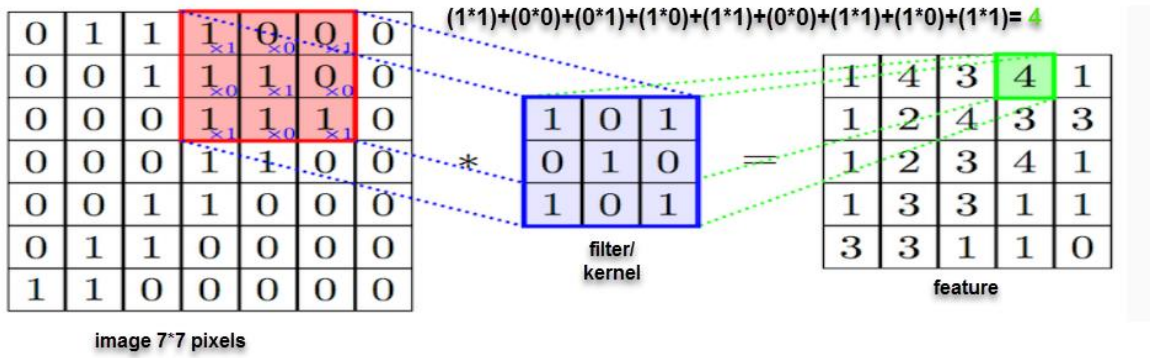

Figure 2.10- Explanation of the convolution operation [Web-6]

B). Pooling layer:

To refine patterns of a convolutional layer, there are pooling layers. A pooling layer is commonly located behind a convolutional layer. The function of a pooling layer is to reduce continuously the dimensionality, which consequently reduces the number of parameters and computations in the network. It also uses a window that slides over the convolution layer results and aggregates them by, for example, selecting only the strongest value or the average of all values in each window. In most cases, a max-pooling layer is used to decreases the feature map size by only keeping the significant information [37].

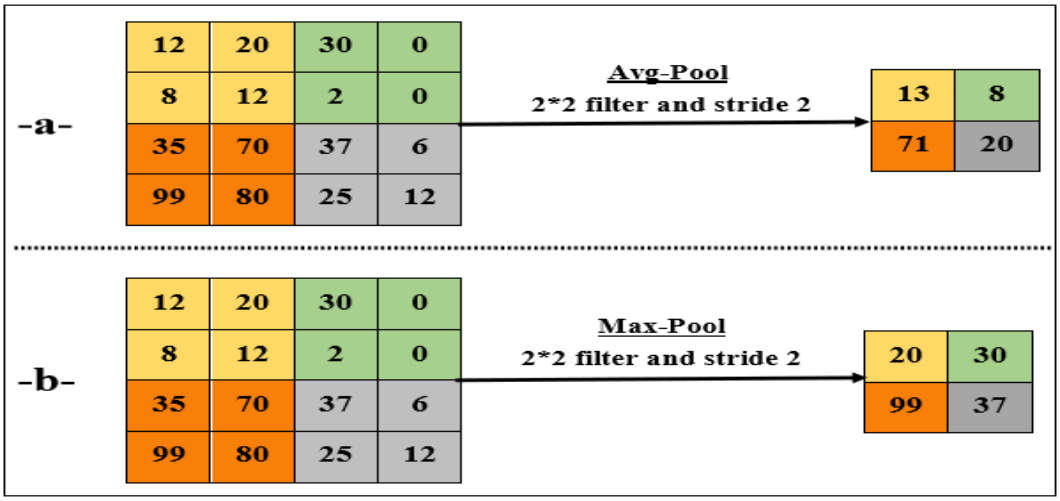

Figure 2.11- Example of pooling layer (-a-: Average-pooling and -b-: Max-pooling)

C). Fully Connected layer:

It is the final layer of the convolutional network can perform highly specific classification by combining all the specific features detected by the previous layers in the input data,this features are generated to 1D feature vector from the 2D features vector (called Flattening). Fully connected layers carry out the same tasks as regular (standard) NNs and attempting to create class scores from activations for classification [38].

5.2. Recurrent Neural Networks (RNNs)

Recurrent neural networks or RNNs [39] are a family of neural networks for processing sequential data .It has the particularity of being able to save its past in order to use it for future predictions (see Figure 2.12). To do this, the RNNs has an internal state (which plays the role of a memory) in which each output is recorded. So the $h_t$output of the current state (decision) is based on the past h_{t-1}output(s).

So the formula of the current state is therefore represented as follows:

$$h_t = f(h_{t-1}, x_t) \tag{10}$$

Applying the activation function TanH:

$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t) \tag{11}$$

Now, once the current state is calculated, we can calculate the output state as:

$$y_t = W_{yh} h_t \tag{12}$$

Where:

x_t is the current input ; h_{t-1} is the previous state ; W_{hh} , W_{hx}, W_{yh} are the weights at previous hidden state,current input state and the output state in ordre respectivly.

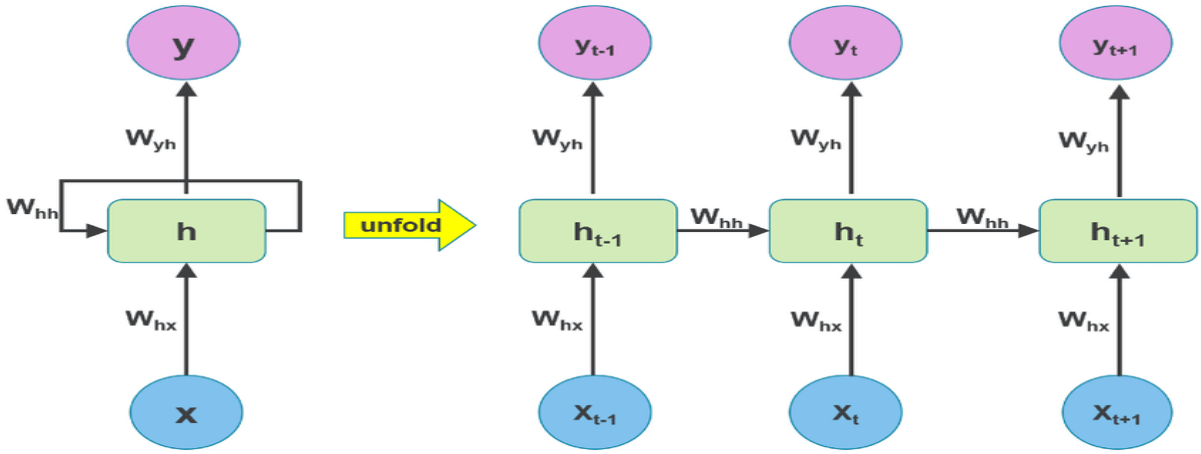

Figure 2.12- Structure of simple (RNN) & unfolded RNN [39]

5.2.1. Types of RNNs

The recurrent neural network can take several types depending on its application [Web-7]:

- **One to one:** it is a traditional neural network.
- **One to many:** uses for example in Music generation.
- **Many to one:** uses for example in sentiment analysis or emotion detection.
- **Many to many:** there are two type in this architecture, (a) and (b) like we see in (figure 2.13). (a) Uses for example in Machine Translation or speech recognition, and (b) in Name Entity Recognition for example.

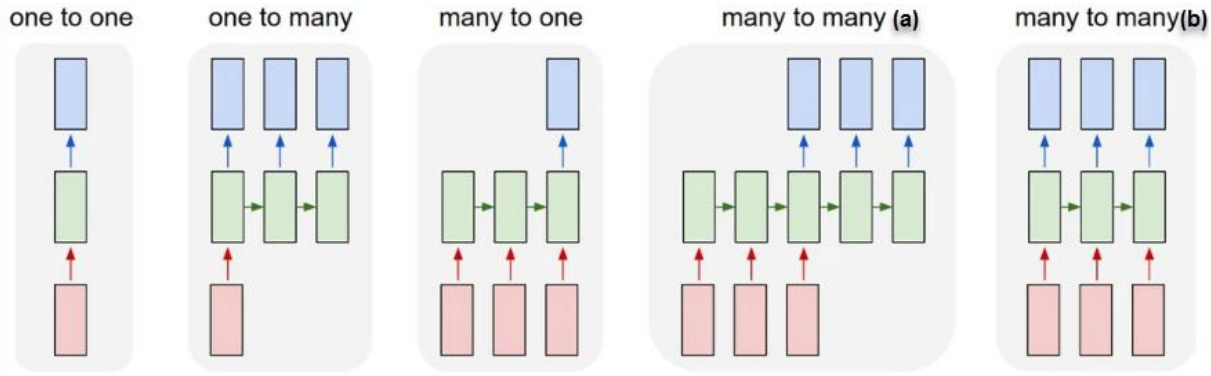

Figure 2.13- Different types of the RNN

5.2.2. Challenges Faced by RNNs

The vanilla RNNs have problems like vanishing gradient which means that the gradient tends towards zero because of the multiplication of the derivatives of tanh, or exploding

gradient (due to multiplication of values that are too large); long-term dependency problem where information rapidly gets lost over time [40].

In most of the real-world problems, variants of RNNs such as LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) are used, which solve the limitations of plain RNN and also have the ability to handle sequential data better [Web-7]. We will try to understand what happens in LSTM in the next section.

5.2.3. Long Short Term Memory (LSTM)

Long short-term memory networks generally known simply LSTM are a special type of RNNs, they were introduced by Hochreiter & Schmidhuber (1997) [41]. This special type were designed specifically to overcome the long-term dependency problem faced by recurrent neural networks RNNs (due to the vanishing gradient problem).

LSTM networks introduce a memory cell that is able to preserve states over long periods of time, overcoming the long-distance dependencies problem of RNNs, The core of the LSTM is a memory cell, which is recurrently connected to itself. It has three multiplication units: an input gate, a forget gate, and an output gate. These gating vectors are in [0,1]. The cell makes selective decisions about what information is preserved, and when to allow access to units, via gates that open and close [41] .The following figure shows a comparison between an RNN cell and an LSTM cell:

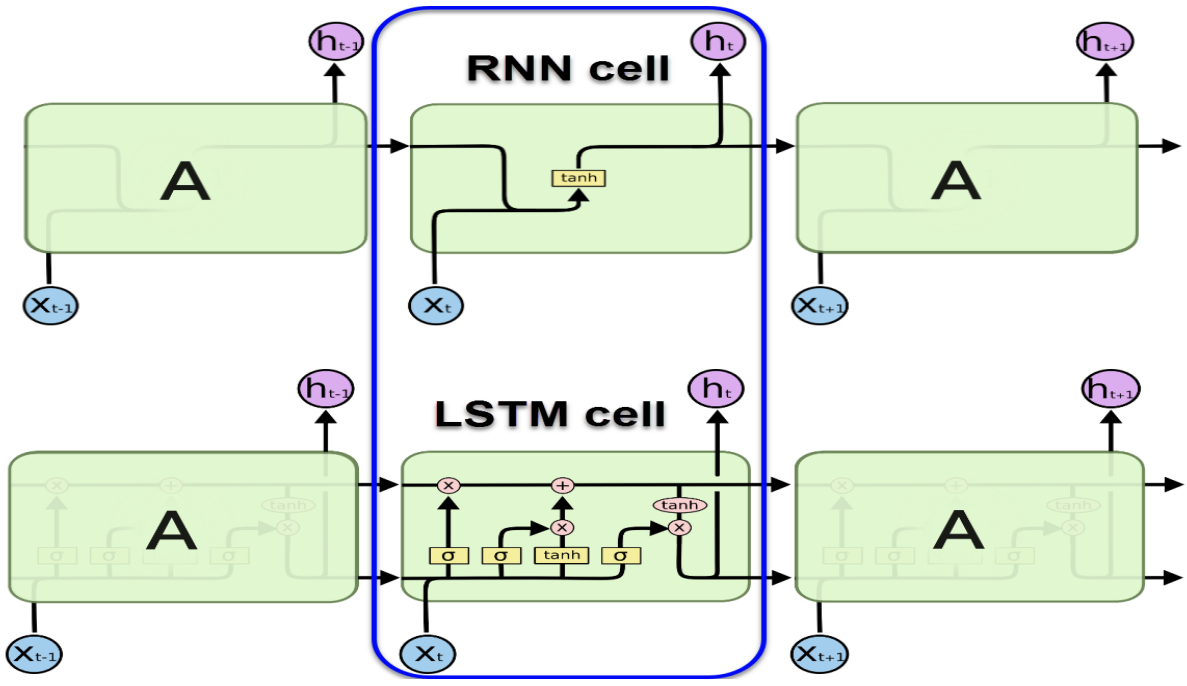

Figure 2.14- Comparison between an RNN cell and an LSTM cell [Web-8]

- **Forget gate:** This gate serves to decide which information should have more attention and which to be ignored.The information from the current input x_t and previous hidden state h_{t-1} are passed through the sigmoid function. the forget gate f_t calculated as follows :

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (13)$$

Where:

W_f : Weight matrix between forget gate and input gate.

b_f : The bias.

The result f_t is a vector whose values are between 0 and 1, Closer to 0 means to forget and closer to 1 means to keep the information. The architecture of this gate is illustrated in the following figure.

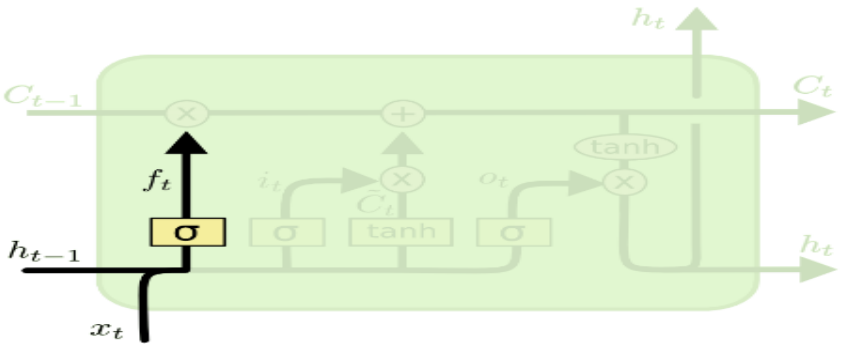

Figure 2.15- Representation of the forget gate [Web-8]

- **Input gate:** The purpose of this step is to add useful information to the cell state. First, the information is filtered using a sigmoid function in the same way as the previous step, using the inputs h_{t-1} and x_t. Then the same inputs are passed to the tanh function, which generates a vector $\tilde{C}_t$ with all the possible values between -1 and 1. The calculations are done as follows :

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (14)$$

$$\tilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \quad (15)$$

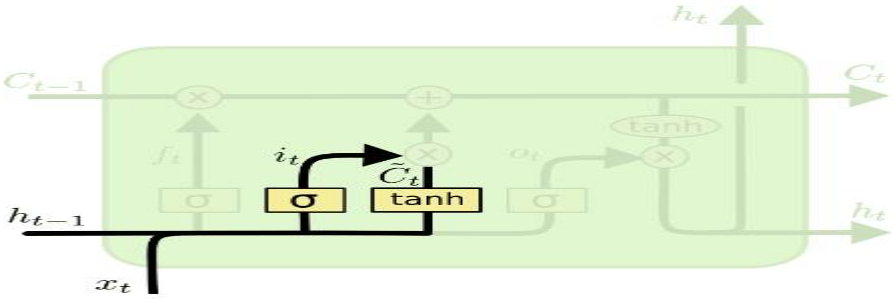

Figure 2.16- Representation of the input gate [Web-8]

To update the old cell state C_{t-1} into the new cell C_t. We multiply the previous cell state by the forget vector f_t then we add the resulte of input gate i_t multiply by the vector of values $\tilde{C}_t$, like the follow equation:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{16}$$

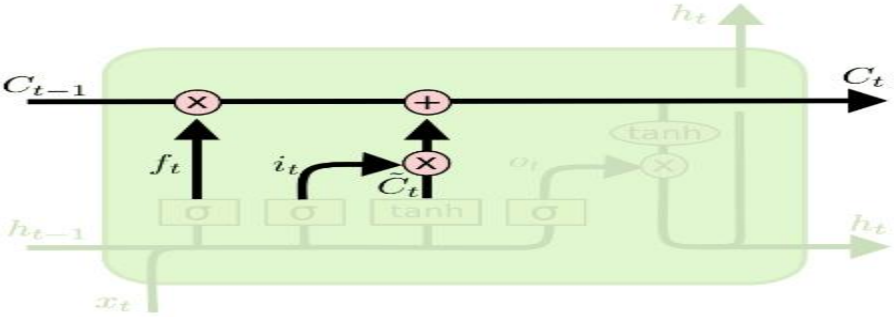

Figure 2.17- Cell state operation [Web-8]

- **Output gate :** This gate aims to determine the value of the next hidden state h_t which is the output of the cell LSTM by the following function

$$h_t = O_t * \tanh(C_t) \tag{17}$$

With O_t: is the output of the function sigmoid, which is equal to :

$$O_t = \sigma(W_O.[h_{t-1}, x_t] + b_O) \tag{18}$$

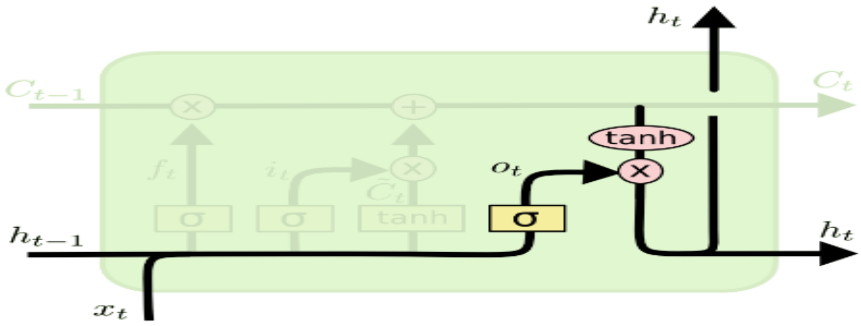

Figure 2.18- Representation of the output gate [Web-8]

6. The Transformer

The Transformer is a sequence-to-sequence type model (Seq2seq), it was introduced in 2017 by Vaswani et al [42] in the article: « **Attention is All You Need »**. Unlike the model in front of 2017 or encoder processing and the decoder, which was carried out with a recurrent neural network, the Transformer is based solely on the **attention mechanism** to ensure the interdependence of words. It uses an encoder-decoder architecture (Figure 2.19). The left component is the encoder, and the right one is the decoder component. Each component has six identical layers, where each layer is composed of sub-layers.

The encoder has two sub-layers: a self-attention **(**Multi-Head Attention**)** is the central element of the Transformer architecture, its role is to maintain the interdependence of words in the representation of the input sequences, and a Feed-Forward Neural Network applied to each attention vector to prepare it for the next encoder. Between each sub-layer, there is a residual connection followed by a layer normalization.The residual connection prevents the vanishing gradient problem, while the normalization prevents the values from changing too much, allowing faster training and better generalization.

For the decoder, it has the same architecture as the encoder with an additional layer Multi-Head Attention (called Encoder-Decoder Attention) placed between the self-attention layer and the Feed-Forward Neural Network, and also modify the self-attention by adding a mask.

The input to the first encoder consists of the word-embedding sum of the input sequence and the vector of the positional encoding. These vectors are passed to the self-attention layer and then in the forward propagation neural networks, the output of each encoder block is used as input to the next encoder until the last encoder which is used by each layer" Encoder-Decoder Attention" in the decoder.

The operation of the decoder is similar to that of the encoder, but with the addition of attention to the representation passed by the last encoder allows it to focus on the relevant information of the input sequence. The output of the last decoder is followed by a linear layer which converts the vector of the output into another vector with the size of the vocabulary of a model. This vector passed to the Softmax layer which transforms the values into probability, the word with the highest probability is added to the output sequence.

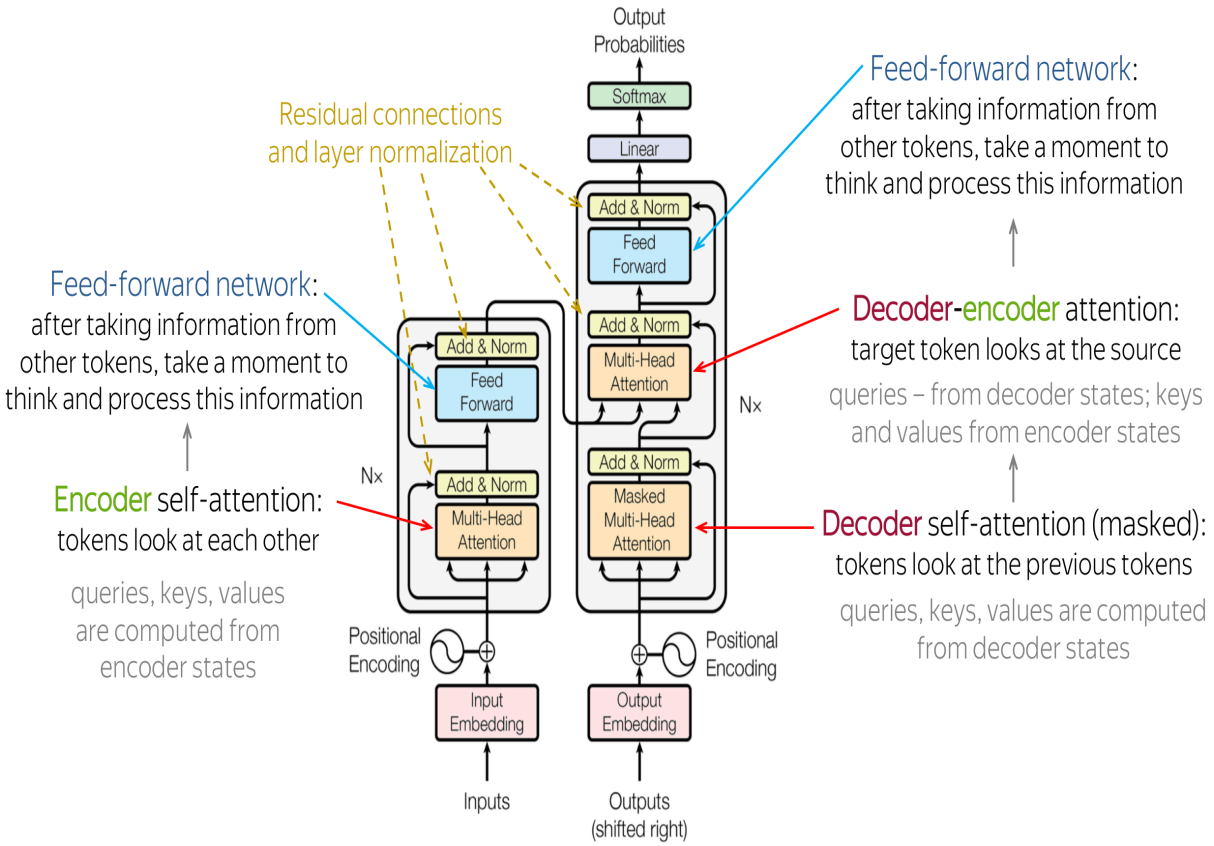

Figure 2.19- Architecture of Transformer [42] [Web-9]

6.1. Transformer detail Functioning

- **Input Embedding:** The input embedding sub-layer converts the input tokens to vectors (each sentence presented with a vector and each dimension model vector equal to 512), using learned embeddings in the original Transformer model [42], the key idea similar words should have similar representation vectors.
- **Positional Encoding (PE):** Positional encodings represent the word's position in the original text as a vector.There are two kinds of PE: Static and learned. The transformer use static PE consists of n sine and cosine waves with different wavelengths and is calculated as presented in Equations :

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (19)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (20)$$

Where: *pos* is the position in time and *i* the features dimension.

The Transformer combines the word vector embeddings and positional encodings. Then it sends the combination results to various encoders followed by decoders [42].

- **Multi-Head Attention:** To understand how a Multi-Head Attention layer works, we must first understand the Scaled Dot-Product Attention layer.

The Scaled Dot-Product Attention head has a simple structure (Figure 2.20), it applies a linear transformation to its query, key, and value vectors. First, calculates the attention score by computing the dot product between the query (Q) and the transposer of key (K^T) vectors, then divide the result by the square root of the dimension of the key vector (d_k) like a scaling factor. Second we uses a softmax function to normalize the resulte (called attention weights), then uses it to weight the values and sum them up to obtain the new input embedding.The key,query and value vectors are the result of the multiplication of the word embeddings input by three weights matrices W^k, W^q, W^v.The equation for the attention matrix output is as follows :

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{21}$$

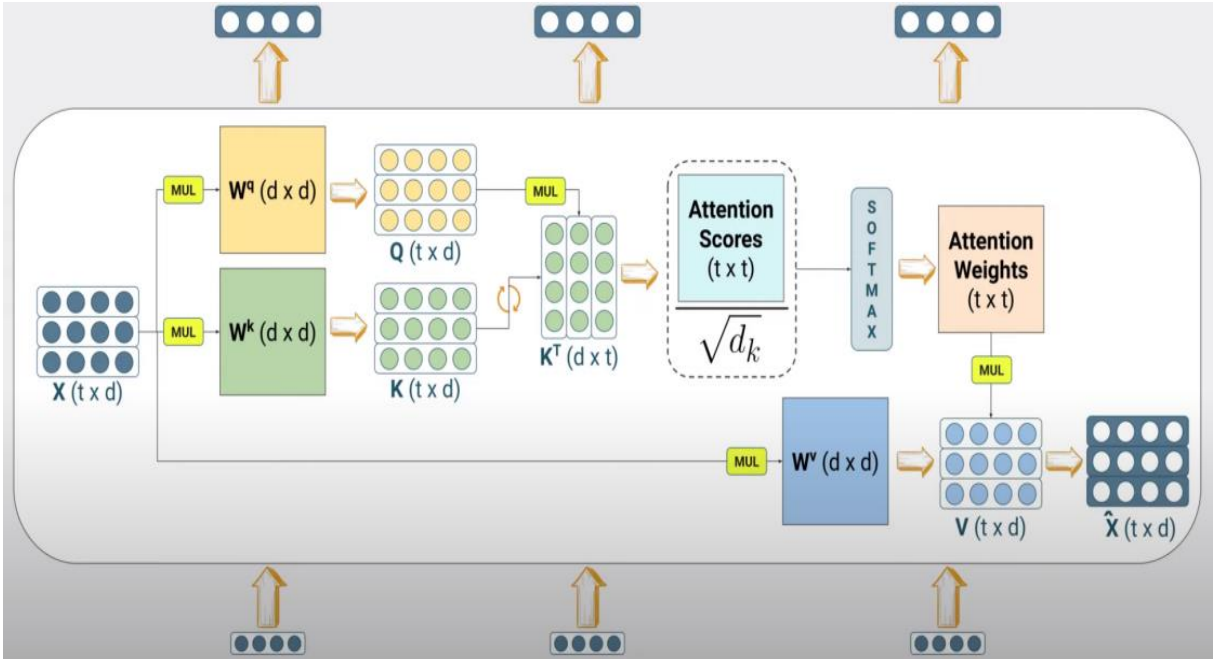

Figure 2.20- Scaled Dot-Product Attention [Web-10]

Like its name, the Multi-Head Attention layer is multiple attention heads. Each attention head will learn a different relationship between the input words and will do so at the same time in parallel (see Figure 2.21). The equation of the output of multi-head attention is:

$$MultiHead(Q,K,V) = Concat(head_0, \dots, head_{h-1})W^O \tag{22}$$

Where:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{23}$$

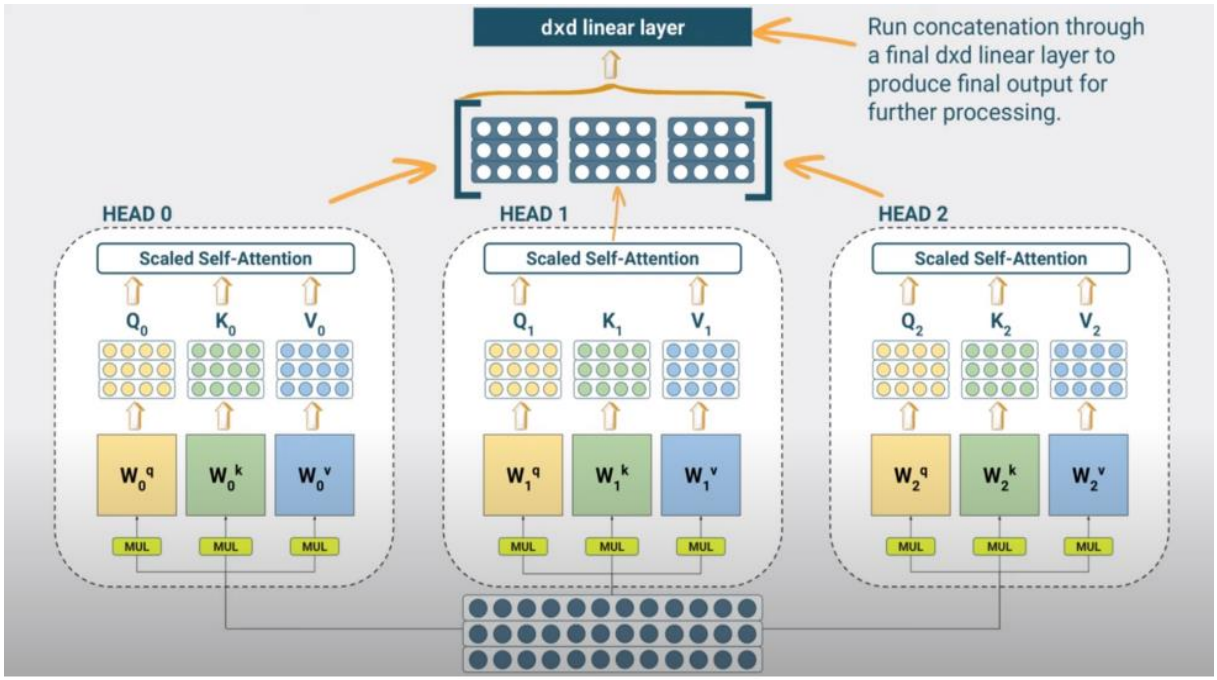

Figure 2.21- Multi-Head Attention consists of 3 attention layers [Web-10]

- **Feed-forward network (FFN):** It is a fully connected network applied to each position independently and identically. This sub-layer is a two-layer feed-forward network with a ReLU activation function. The first layer is four times the size of the model (d_{ff} equal to 2048). This seems to give the transformer enough representational capacity. The second layer will project the output of this first layer into the original size (d_{model} equal to 512). Given a sequence of vectors $h_1,\ldots, h_n$ the computation of a position-wise FFN sub-layer on any h_i is defined as shown in the equation 23:

$$FNN(h_i) = ReLU(h_i W_1 + b_1) W_2 + b_2 \quad (24)$$

- **Masked Multi-Head Attention:** multi-head where some values are masked (i.e, probabilities of masked values are nullified to prevent them from being selected).Whene decoding, an output value should only depend on previous outputs (not next outputs). Hence we mask future outputs. The equation of the output of masked attention is :

$$maskedAttention(Q, K, V) = softmax\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V \quad (25)$$

Where: M is a mask matrix of 0's and -∞'s

6.2. Different Transformer models:

The emergence of pre-trained models has ushered Natural Language Processing (NLP) into a new era, they not only allow amateurs to perform simple NLP tasks, but also help experts achieve better results without creating a model from scratch.

Pre-trained models based on the Transformer architecture.They are pretrained on a massive amount of unlabeled data in order to learn the universal representations of a language, and then the models can be refined on specific labeled datasets to the targeted task, which leads to better performance compared to training on label data alone. There are many pre-trained language models, they differ according to the architecture (Encoder, Decode, both encoder-decoder components), the preprocessing dataset, the training objective, the number of hyper-parameters. The most used are BERT (Bidirectional Encoder Representations from Transformers) [43], T5 (Text-To-Text Transfer Transformer) [44] and GPT (Generative Pre-trained Transformers) [45], the following table summarizes a little of different architectures of the pre-trained models with their corresponding tasks.

Table 2.1- Comparative overview of pre-trained models

Language models	**Architecture**	**Tasks**
BERT - Google (2018) -	Encoder	Named Entity Recognition (NER) Classification of sentences.
GPT - OpenAI(2018) -	Decoder	Texts generating.
T5- Google (2019) - BART- Facebook (2019) -	Encoder-Decoder	All NLP tasks.

7. BERT

BERT [43], stands for Bidirectional Encoder Representations from Transformers. It is a pre-trained language model built on top of the Transformer blocks, came out of Google AIlabs in late 2018. The BERT model architecture is a stack of encoder transformer architecture, this language model is bidirectional, which means it learns information from left to right and from right to left, this feature allows it to have a better understanding of text, it is consist of two models. The first is BERTbase composed of 12 layers with 768 hidden sizes and 12 self-attention heads with Total Parameters = 110M. The second is BERTlarge composed of 24 layers

with 1028 hidden sizes and 16 self-attention with Total Parameters = 340M [43]. The pre-training of this model is done on two tasks:

- **Masked language modelling (MLM):** This technique is used to mask 15% of the words randomly in a sentence and the model tries to predict the masked word based upon the context of the other (non-masked) words in the sentence.
- **Next sentence prediction (NSP):** In this task the model is fed with two sentences A and B as an input and is supposed to predict if the second sentence (B) is the subsequent sentence in the document. More precisely, 50% of the time B being the sentence that follows A (labeled as IsNext) and 50% being a random sentence (labeled as NotNext) [43].

Both the MLM and NSP tasks are used to model contextual and semantic relationships in the training corpora to build a language model. The language model than can be used in a Transfer Learning context to learn another task upon it. This approach reduces the resource requirements for many NLP tasks since the resource heavy task of learning a language model has to be trained only once. To train the language model, Devlin et al [43], used the BooksCorpus (800M words) and English Wikipedia (2,500M words).

To help the model distinguish sentences in training, the encoder input is processed in three steps before it is passed to the model (see Figure 2.22):

The first step consists of inserting a [CLS] token at the front of the sentence, and a [SEP] token is inserted at the end of each sentence. While the second step adds segments A for the first sentence and B for the second, this allows the encoder to distinguish the order of the sentences. The last is to indicate the position of each token in the sequence.

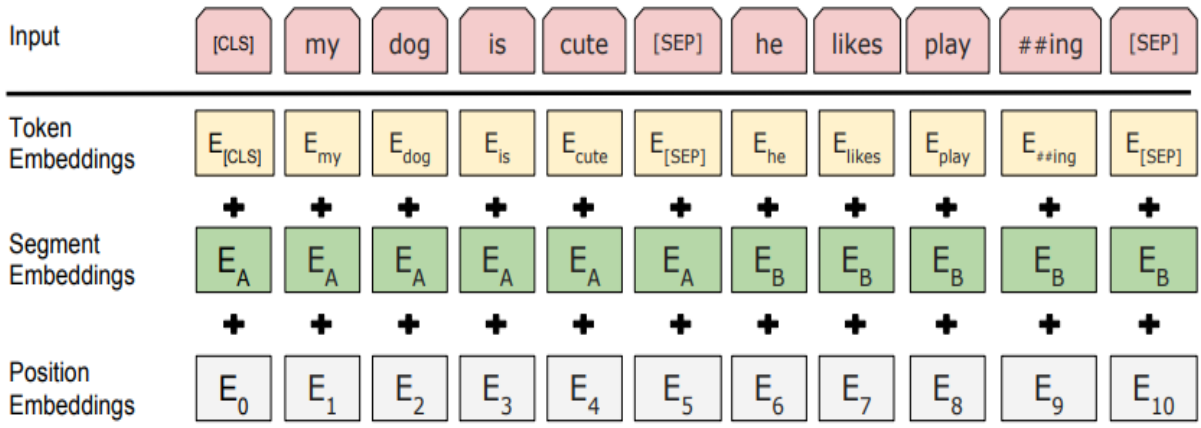

Figure 2.22- BERT input representation [43]

8. Conclusion

Deep Learning is a very rich field in which certain basic concepts are essential to its understanding. In this chapter, we have tried to cover some concepts and algorithms in ML and the majority of existing types of learning. Then we passed to DL and it populare architecture. We have focused in explanation mainly on models dedicated to the processing of sequential data like RNNs, and the Tronsformer.

In the coming chapter, we will present the proposed approach for the Sentiment analysis, as well as the experiments followed and the results obtained.

Chapter 3 : Conception and Experimentation

1. Introduction

In the previous chapters, we have briefly introduced the definitions, and architectures needed for our conception.

The goal of this last chapter is to present our proposed approach for the sentiment analysis based in Algerian dialect. In the first part of this chapter, we present the motivations that we have set ourselves. Then, we approach the various stages for the implementation of our model for sentiment analysis. After that, we present the experiments and the evaluation of the different models built. In the last, we present the tools (software and hardware) used in our work.

2. Motivation

Because of the few works and researches on the classification of sentiments with the Algerian dialect. The goal of this work is to adopt successful approaches from the field of NLP, and based on transfer learning we fine-tune them to the sentiment analysis of Algerian consumers, depending in dataset contain theirs comments in social media. Here, we fine-tune the DziriBERt (see 3.3) architecture for sentiment analysis with its self-attention-based approach. For this, we must first perform a series of experiments in order to determine the hyper-parameters of the model, and the pre-processing in data that provide the best results in terms of classifaction ability of comments to negative, neutral, or positive.

3. General System Architecture (Proposed Approach)

This section describes different stages followed to design the proposed approach, giving a general diagram and then explaining the details of each stage.Basically, our approach consists of three major stages as shown in Figure 3.1: stage (I) for our Dataset (the data collection and annotation), the data preprocessing stage (II), where we preprocess our dataset. And (III) the classification (training and testing) stage where using transfer learning, by selecting pre-trained model and fine-tune this model with our training dataset.

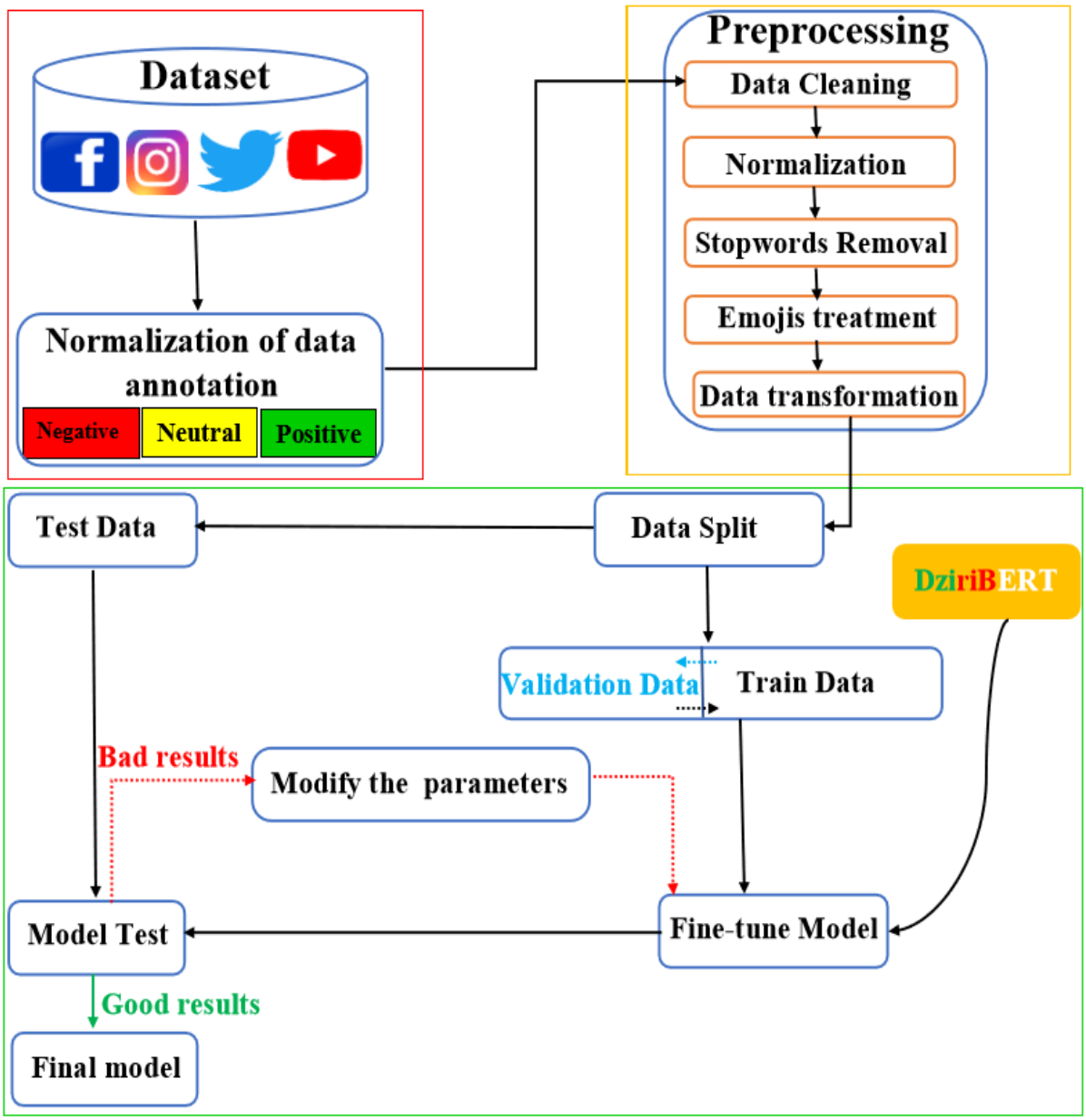

Figure 3.1- Our Proposed Approach

3.1. Dataset

The first step in our Sentiment Analysis process was to collect comments in the Algerian dialect. This was done from two datasets.

- **First Dataset:** This open source dataset from kaggle [Web-11], contains 5320 comments was collected by DJOUGHI Mehdi, graduate student from University of Algiers 1 Benyoucef BENKHEDDA. This dataset collected from two sources, 34% from facebook, and 66% from instagram. This is done by manually collecting comments from the pages of telephone operator Djezzy. The data has been annotated manually by

the author for 3 categories: -1 refers to the negative comment, 0 refers to the neutral comment, and 1 refers to the positive comment.

The figure below represents the distribution of the comments in this dataset, which is 1819 negative (red color), 2934 neutral (yellow color), 567 positive (green color).

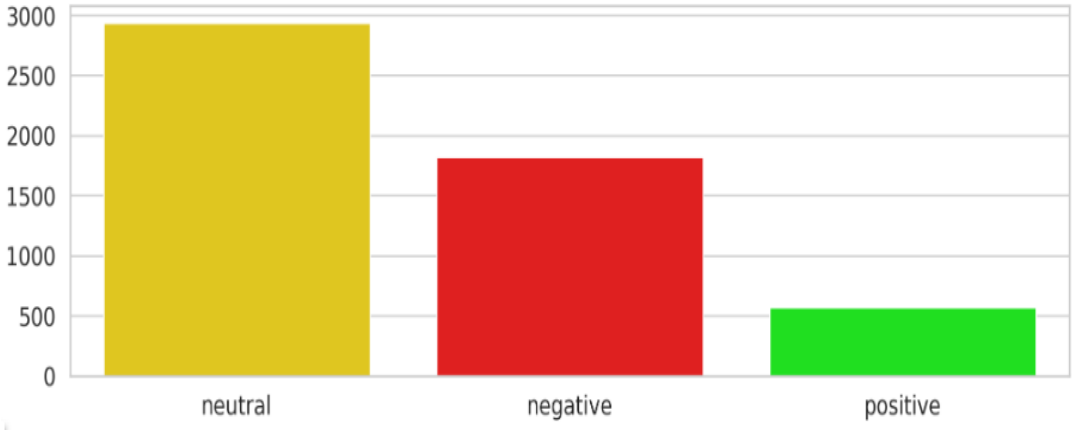

Figure 3.2- Comments distribution from the first Dataset

- **Second Dataset:** This dataset contains 7044 comments was collected by students from university Saad Dahleb Blida 1, from four sources. The two first sources are facebook and instagram, this is done by manually collecting comments from the pages of telephone operator Djezzy, Mobilis, and Ooredoo. The second two sources are Twitter and YouTube. This task was accomplished by extracting comments from Twitter posts, as well as videos posted on YouTube by the presenters from the Algerian telephone operators. This was done using twitter API and youtube API. The data has been annotated manually by their authors for 3 categories (see Figure 3.3) where 1 refers to the negative comment (3094 comments), 0 refers to the neutral comment (2429 comments), and 2 refers to the positive comment (1521 comments).

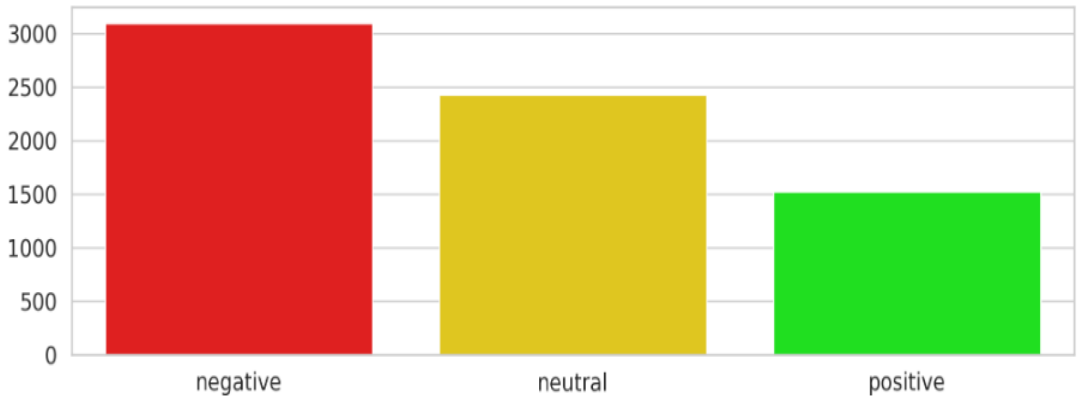

Figure 3.3- Comments distribution from the second Dataset

3.1.1 Normalization of data annotation

In this step, we change the annotation of the first and second datasets, where the annotation of the 3 categories of comments in each dataset look like this : 0 would correspond to a negative comment, 1 would correspond to a neutral comment, and 2 would correspond to a positive comment. After that, we combine these datasets to get one unbalanced dataset with unified annotation of 12364 comments distributed in: the Algerian dialect written in Arabic characters, the Arabizi written in Latin characters, the French, English, and a mixture between French and Arabizi (see table 3.1).

Table 3.1- Comment Annotation Examples

Comment	Annotation
Mauvaise cnx dmg😐	0
عرض امتياز علاش نحتوحه ؟	1
Mliha had l'application bzaaf tsahel ❤️	2

The figure below represents the distribution of our comments, which is 4913 negative, 5363 neutral, 2088 positive.

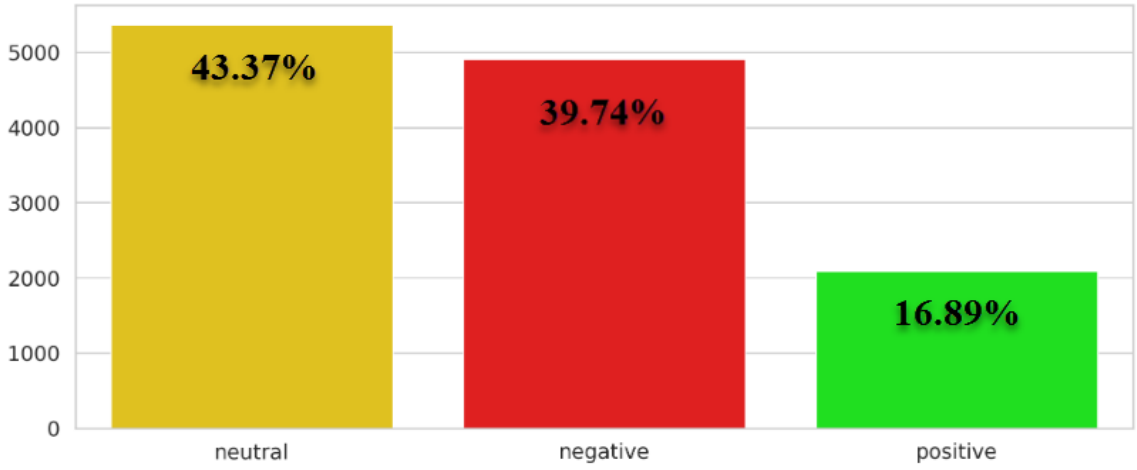

Figure 3.3.4- Comments distribution in the final dataset

3.1.2 Data Exploration

Data exploration is an essential step in the analysis of comments and useful for the realization of the next Stage, as long as it allows us to better understand our corpus and to know the nature and the language with which the comments are written. In this phase, we examined the dataset using the word cloud (See Figure 3.5) that allows to us to have a global view of the text and

words used in our corpus.where each word's size is proportional to its importance (more frequent words appear larger).

Figure 3.5- Word cloud of our dataset

3.2. Data Pre-processing

The pre-processing is the process in our data to reduce errors and noise, and enhance sentiment analysis performance [46]. These techniques are an essential step in the SA for Arabic text. Especially the Algerian dialectal text because of its unstructured form.

3.2.1. Data Cleaning

In this step we reduce the noise by removing all emails, tags (e.g.@username), URLs (http\www), phone numbers, punctuations (e.g.؟ ؟ ! % & ,..etc) , and arabic diacritics(e.g. fatha(ً),Damma (ٌ),Tashdid(ّ), Kasra (ٍ), Tanwin Damm (ٌ), Tanwin Fath (ً),Tanwin Kasr (ٍ), Sukun (ْ), and Tatwil(ـ)), like we showing in the Table 3.2.

For hashtags (#Hash), we decided in this work to take them in consider, and not remove them because potentially contain a sentiments. For example in our data, the hashtag "#merci_Mobilis❤❤" carries positive sentiments.

Table 3.2- Example of data cleaning procces

Comment	Comment after the cleaning process
@ABS_011 الأن مُمْكِن نلعبُو كِيمَا النَّاس ❤❤!!!! mrccccccc YA3tikoum Saha	الأن ممكن نلعبو كيما الناس ❤❤ mrccccccc YA3tikoum Saha

3.2.2. Normalization

This step contains two substeps are widely used in Arabic NLP:

- **Letter normalization:** It aims to unify the letters that can appear in different forms. In this work, we replace {إ أ آ ا} with {ا}, {ة} with {ه}, {ؤ ئ} with {ء}, {ى} with {ي}.Furthermore, all letters of comment are converted to lowercase.
- **Elongation removal:** People may repeat characters for emphasis or showing a strong emotion, or also repeated spaces, especially on social media. In this substep, the word is reduced to its standard form by removing these repeating letters, and deleting all unnecessary spaces. We keep only two repeated consecutive letters or spaces in this work (For example, "بزاااااف هايله" become to "هايله بزااف").

Table 3.3- Example of Normalization procces

Comment	Comment after the Normalization process
الأن ممكن نلعبو كيما الناس ❤️❤️ mrccccccc YA3tikoum Saha	الان ممكن نلعبو كيما الناس ❤️❤️ mrcc ya3tikoum saha

3.2.3. Stopwords Removal

This process involved the elimination of words that just serve to structure, and organize the language but do not contain additional meaning or sentiment to the text. There are several tools and libraries for removing stop words from languages like English, French, and even MSA (Modern Standard Arabic), but none of them provides a list of stop words used in the Algerian dialect. As a result, we manually prepared a list of stop words for the Algerian dialect written with Arabic letters, and others with Latin letters, for example ['3lih', 'win', 'نتا', 'الان', 'نتوما'].

Table 3.4- Example of Stopwords Removal procces

Comment	Comment after the Stopwords removal procces
الان ممكن نلعبو كيما الناس ❤️❤️ mrcc ya3tikoum saha	ممكن نلعبو الناس ❤️❤️ mrcc ya3tikoum saha

3.2.4. Emojis treatment

In this step, we have removed repeated consecutive emojis. Next, we included matching the inputs with a two lists of Unicode emojis, one for the positive emojis contains 34 positive

unicode emojis, and one for negative emojis contains 23 negative ones. In the actual implementation of our system, when an emoji is matched it is replaced by a specific term that is not actually part of the Arabic vocabulary, the term is used to identify if the emoji unicode is positive or negative, where the positive emojis is replaced by "ايموجيموجب", and the negative emojis is replaced by "ايموجيسالب".

Table 3.5- Example of Emojis treatment procces

Comment	**Comment after the Emojis treatment process**
ممكن نلعبو الناس ❤❤ mrcc ya3tikoum saha	ايموجيموجب ممكن نلعبو الناس mrcc ya3tikoum saha

After the first four steps of pre-processing (Data Cleaning, Normalization, Stopwords Removal, and Emojis treatment), the word cloud of our dataset looks like we show in the figure below.

Figure 3.6- Word cloud of our dataset after the pre-processing

3.2.5. Data Transformation

This step consists of converting data to correspond to the fine-tuned DziriBERT model (see Figure 3.7). First, we used DziriBERT Tokenizer to decompose the sentences into a set of meaningful pieces "tokens" (tokenization), because this tokenizer is the most relevant for Algerian dialect. Seconde, the special token [CLS] symbol is added at the start of every sentence and the token [SEP] added in the end of each sentence, after that, we get BERT's format. Third,

we Padding ([PAD]) and truncate all sentences to a single constant length, this Depending on number of tokens in the most comments in our dataset. Then based on the vocabulary of pre-trained DziriBERT, each token is mapped to an index. Finally, we create an attention mask; list of 0 and 1 indicating whether the model should consider the tokens or not when learning their contextual representation. We expect [PAD] tokens to have value 0.

Figure 3.7- Data transformation procces

3.3. Classification

In the proposed approach, sentiment classifier will be created by selecting DziriBERT like a pretrained model for fine-tune it with our data.

DziriBERT is pretrained language model uses the same architecture of BERTbase (12 encoders, 12 attention heads, and a hidden dimension of 768), it was performed specifically for the Algerian dialect. The training set of this pretrained model containe 1.2 Million tweets that were posted from major Algerian cities. It handles Algerian text contents written using both Arabic and Latin characters [47].

The fine-tuning of DziriBERT is is the same as that of BERT For sentiment analysis, or other Multi-label classification problems, a linear (fully-connected) layer with a standard softmax activation function is added to the last hidden state of the first token (the [CLS] token) as shown in Figure 3.8. With a hidden state vector C $\in R^H$ where H is the dimension of the hidden state and a fully-connected classification layer with weights W $\in R^{K\times H}$ where K is the number of classification labels, the label probability after applying the softmax function is then $P = softmax(CW^T)$ [43][47].

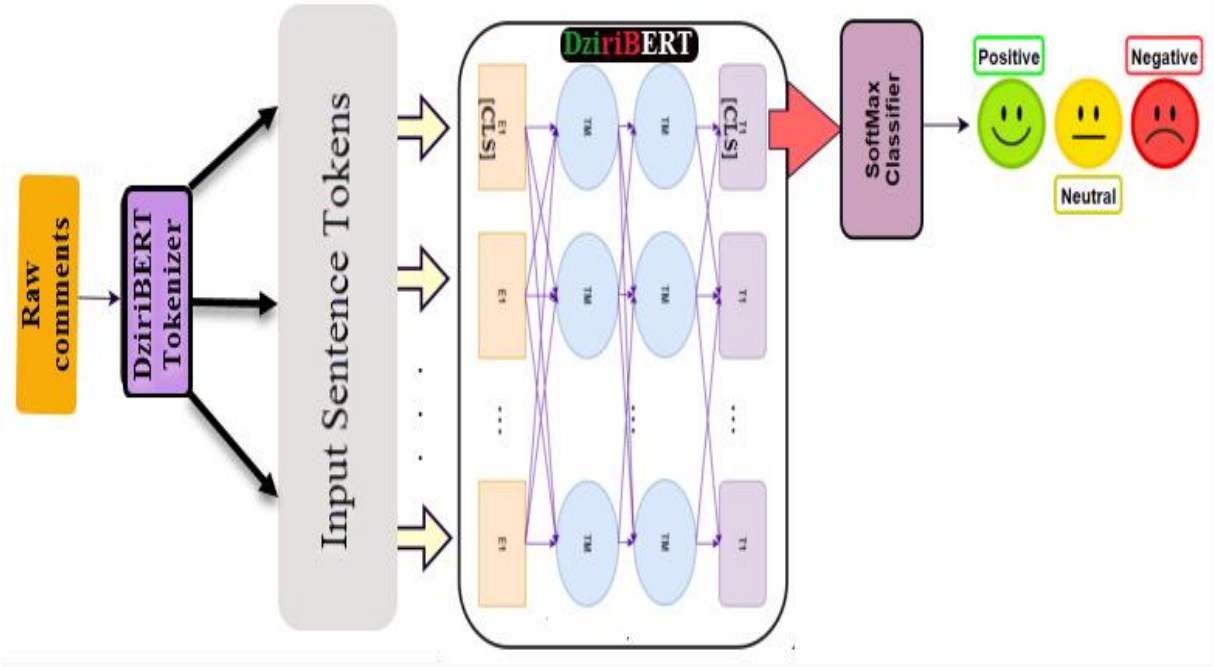

Figure 3.8- DziriBERT Fine-Tuning Model Architecture for Sentiment Analysis

4. Experiments, Results and Discussion

It is generally considered that it is difficult to create an ideal model the first time, and that it takes trial and experimentation to find the optimal values. When learning (training) our model, the major problem we had to overcome was the Overfitting. We therefore resorted to certain regularization tips and techniques.

In this section, we have detailed the experimental used techniques and some of the results obtained using the proposed approach. The experiments aimed to solve the problem of overfitting, explore the capabilities of the model, and identify the hyper-parameters, and the pre-processings procces that provide the best results.

4.1. Number of epochs and Batch-size

An epoch refers to one cycle through the full training dataset. The number of epochs must be defined carefully, because it is very important in determining the efficiency of the model. Following the work done in [47], in this work the model DziriBERT have been fine-tuned for three epochs.

As for the batch, it is specifies how many samples must be processed before the internal model parameters are updated [Web-12]. In the training, the data can be divided into one or more batches. To train the model and test it in our work, we set the batch size to 16.

4.2. Cross-Validation

Cross-validation is a technique that simultaneously monitors and prevents overfitting. It consists allocating a portion of the training set to conducting a test step at every epoch. Thus, it is possible to keep track of the model performance on unknown data to determine how effectively it generalizes. In this way, a good visualization of the performances makes it possible to determine more or less the source of the problem and to move towards a suitable solution. To perform cross-validation during training. First, we splitting our dataset to 20% for testing and 80% for training. Then we split the training set in eight folds (seven folds for training and the remainder for validation).

4.3. Experiments on Dropout

Certain neurons with weights that are too low relative to the others are not taken into account during training, which leads to increase the generalization error (problem of the ovefitting). To avoid this, we using a dropout technique, which can be applied at the exit of some of the network layers. It will randomly and periodically remove some of the neurons (along with their input and output connections) from the network in the training of the model.

This experiment consists to finding the rate of dropout regularization to use to reduce overfitting and improve the generalization of our fine-tune model, where works well both on the training set and on the new data (test set). The results of this experiment for different dropout rates are shown in Table 3.6.

Table 3.6- Experimental Results to Determine Dropout Rate

Dropout rate %	**Accuracy in training set %**	**Accuracy in test set %**
10	94.93	80,75
15	94.88	81.15
20	**94.88**	**81.88**
25	95.03	81.23
30	94.97	81.07

From the table above, it can be seen that the best dropout rate from the experimental results for our model is equale to 20%, where it is give the best accuracy 81.88% on the test set, and whene we increase or decrease the dropout rate from this value, the accuracy begins to decrease.

4.4. Experiments on Learning rate (Lr)

Choosing the learning rate that is tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function, is difficult. Because too low a value can lead to a long training process that may be stuck, on the other hand too high a value can lead to learning a suboptimal set of weights too quickly or an unstable training process. When train a model, the learning rate may be the most important hyper-parameter to consider.

Beceause DziriBERT have same achitecture of BERT, we did experiments on learning rates that The BERT authors recommended for fine-tuning the model (see Table 3.7).

Table 3.7- Experimental Results to Determine Learning Rate

Learning rate	Accuracy in training set %	Accuracy in test set %
1e-5	91.39	81.43
2e-5	**94.88**	**81.88**
3e-5	95.88	80.79
5e-5	95.89	80.26

From Table 3.7, it seems clear to us that the best value for the learning rate is 0.00002 (2e-5) with accuracy of 81.88% in test set, as other values for this hyper-parameter decrease the efficiency of the model.

4.5. Experiments on Pre-processing

The pre-processing has a significant impact on building a good model. A wrong pre-processing or not appropriate to the type and structure of the data that we are dealing with will inevitably lead to a decrease in the efficiency of the model, whether in the training or testing.

In these experiments, we want to ensure that the pre-processing operations on emojis and hashtags are appropriate for our data.

Table 3.8- Experimental Results on Emojis

Emojis	**Accuracy in training set %**	**Accuracy in test set %**
Treatment	**94.88**	**81.88**
Remove	92.73	78.20

Table 3.9- Experimental Results on Hashtags

Hashtags	**Accuracy in training set %**	**Accuracy in test set %**
Take in consider	**94.88**	**81.88**
Remove	94.96	81.03

From the Table 3.8, we note that when all emojis were removed, the accuracy of the model on the test set decreased by 3.68%, compared to when they were removed repeated consecutive emojis and then replaced with words that express them. Whereas, the model gave an accuracy of 81,88% in test set when replacing emojis with expressive words and 78,20 % when removing the emojis completely.

After that, from Table 3.9, we can clearly notice that the accuracy of the model in test set when removing the hashtags decreased from 81.88% to 81.03%, and this confirms our belief that the hashtags carries sentiments in the data that we are working on.

4.6. Experiments on Max-length of sentence tokens:

In our work, in order to operate on a batch for training or test the model, we need to truncate and pad all of our input sequences to be a single fixed length. To chois the best length of sentences (number of tokens) in our data set, we will make some experement (see Table 3.10).

Table 3.10- Experimental Results on Max-length

Max-length	Accuracy in training set %	Accuracy in test set %
50 tokens	94.90	80.91
65 tokens	**94.88**	**81.88**
75 tokens	95.05	81.80

From the table we note that the best accuracy of the model in the test set 81.88% was when we set the max_length value to 65 tokens. But when max_length is set to 50 or 75 tokens, the accuracy of the model is reduced somewhat. This decrease in the accuracy of the model can be explained by:

- When choosing max-length equal to 50, there will be a loss of important information and sentiments contained in the tokens after fifty. Because there is a significant percentage of our sentences that carry a number of tokens greater than the max-length we set (see Figure 3.9).
- On the other hand, when max_length is set to 70, there is no significant loss of important information and sentiment contained in the tokens. But there are a large number of insegnificant tokens (padding) that are used as padding for short sentences like we shown in Figure 3.9. These padding tokens will inevitably affect the classification report of the model and thus lower the accuracy of the model.

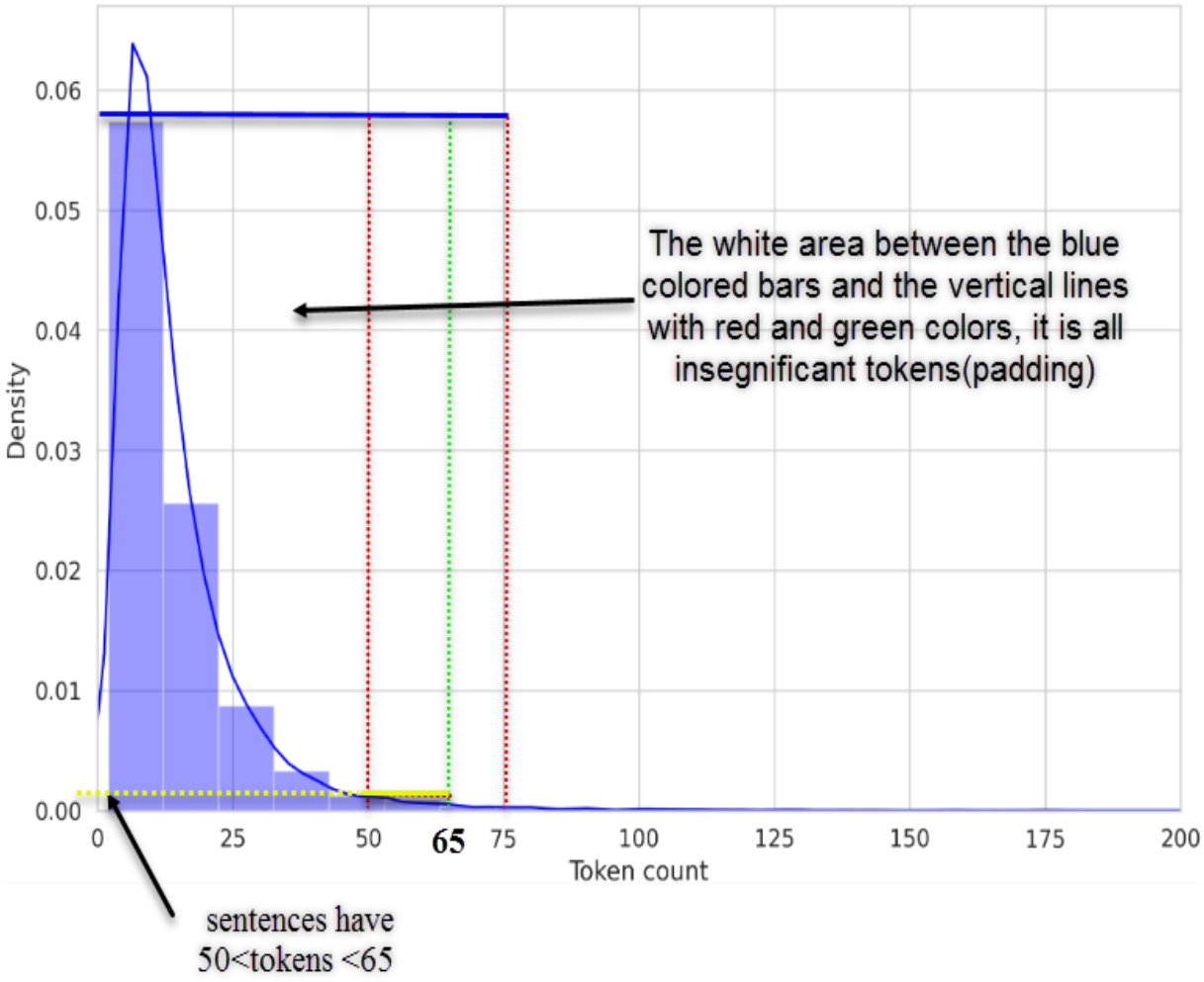

Figure 3.9- Number of tokens per comment

5. Evaluation of Performances

5.1. Evaluation matrices

In order to evaluate the fine-tuned model and determine if the classifers are accurate in capturing and predicting a pattern, we needed to assess the performance of the model. A confusion matrix was used as an indicator for the accuracy of the classifer results, in order to help to obtain a better analysis result. In general, this matrix was about measuring the learning algorithm accuracy on a test/labeled dataset. The positive class was“yes” and negative class was“no”, where P denotes the number of positive classes and N the number of negative classes. Additionally, there were diferent terms that are used in the confusion matrix which were as follows [48]:

- **True positives (TP)** refer to correct classifcations labeled “positive”.
- **True negatives (TN)** refer to correct classifcations labeled negative “labeled”.

- **False positives (FP)** refer to incorrect classifcations where the outcome is the predicted class “yes”, but the actual class is “no”.
- **False negatives (FN)** refer to incorrect classifcations where the outcome is the predicted class “no”, but the actual class is “yes”.
- Moreover, from the confusion matrices we computed a list of rates such as the following [49] :

❖ **Precision** identifes the success degree of the classifers in correctly predicting the number of labeling dataset and the total number of labels in the test dataset that were correctly predicted as positives.

Precision = TP/TP+FP

❖ **Recall** shows the success degree of the classifers in correctly predicting a number “ratio” of true positive-labeled dataset and a total number of positives and negative labels in a test dataset.

Recall = TP/TP+FN

❖ **F1-score** is a collection of the precision and the recall that gives the total overview of the measured performance of the classifer.

F1-score = 2(Recall * Precision) / (Recall + Precision)*

❖ **Accuracy** describes the degree of how the classifer classifed and predicted documents correctly from the total number of all documents in a test set.

Accuracy = TP+TN/TP+FP+FN+TN

❖ **Support** shows the number of the true response samples of the class in the dataset.

❖ **Macro avg** computes the average precision, recall, and F1 of all classes.

❖ **Weighted avg** calculates the precision of all classes merged together.

Weighted average = (TP of class 0 + TP of class 1) / (total number of class 0 + total number of class 1)

5.2. Evaluate the model performance

Through the experiments that we performed, we chose the hyper-paremeters shown in Table 3.11, to be implemented in the development of our model.

Table 3.11- Hyper-parameters used in the approach

Hyper-parameters	Value
- Batch-size	**16**
- Epochs	**3**
- Dropout	**0.2**
- Max length	**65**
- Learning rate(Lr)	**2e-5**
- Optimizer	**AdamW**
- Loss function	**CrossEntropy**

After training, the model with the hyper-parameters seen in the Table above, we tested it on 20% of the total dataset. The obtained results are shown in Figure 3.10.

-a- Confusion matrix

-b- Classification report

Figure 3.10- Model test results (-a- Confusion matrix, -b- Classification report)

From the Confusion matrix shown in the Figure 3.10. On the diagonal, it can be seen that the negative sentiments is the best predicted class with a precision of 83%, followed by the

neutral sentiments with a precision of 82%, and finally the positive sentiments have the lowest with a precision of 80%.

On the other hand, from the outliers of the matrix, we notice that there is confusion between the different classes of sentiment and specially the neutral sentiment with the negative, also between the neutral and positive. This is not surprising since this confusion is presumably due to the use of negative or positive words and emojis to express neutral sentiment. In addition to some other problems for example sarcastic sentences, domain considerations, and thwarted expectations. The Table 3.12 show some examples of these problems in our data.

Table 3.12- Example of some false predictions of our model

Comment	**True Sentiment**	**Predicated sentiment**	**Problem**
ايموجيسالب ما عنديش الزهر	Neutral	Negative	Use a negative emoji to express neutral sentiment
مرحبا بيكم لافامي ايموجيموجب	Neutral	Positive	Use a positive emoji to express neutral sentiment
عروض مليح والريزو مهوش كامل لبلايص مليح	Positive	Negative	Thwarted expectations
علاه متحبسوس وتقلبوها محاجب	Negative	Neutral	Sarcastic

Some times, a human or even an expert can also fall into the confusion of categorizing the sentiment in the comments into positive and neutral, or negative and neutral, and this is what makes the error of the classifier logical.

From the classification report, we can see that the accuracy of our model in the test set is equal to 81.88%, and this is very acceptable given the lack of works on the Algerian dialect, especially the classification of consumer sentiments like in our case.

5.3. Comparison

As part of this work, the proposed model is compared with the work of [23], in terms of precision, recall, f1-score for each one of our selected sentiments (negative, neutral, or positive). The comparison is shown in Table 3.13.

The work [23] propose an approach to analyze the sentiments of the Algerian dialect for the benefit of the Algerian Telephone Operator Ooredoo. The proposed approach is based ona CNN deep learning model and the SVM machine learning algorithm. Their corpus contains 65,125 comments, where the Algerian dialect is widely used with 75% of the collected data, while the rest of the dataset consists of other languages.

Table 3.13- Comparison results

Evalution measure	Model	Negative	Neutral	Positive
Precision	CNN [23]	72%	70%	76%
	SVM [23]	71%	64%	72%
	Our model	**83%**	**82%**	**80%**
Recall	CNN [23]	81%	73%	37%
	SVM [23]	77%	74%	24%
	Our model	**84%**	**81%**	**80%**
F1-score	CNN [23]	73%	68%	40%
	SVM [23]	72%	69%	29%
	Our model	**83%**	**81%**	**80%**

From the table of comparison above, it is remarkable that our model gave the best results and with a good percentage, compared to the CNN and SVM models of [23] in the evaluations mesures used to evaluate the three classes of sentiment. Although the data volume used by [23] is much larger than ours data (about 5 or greater 6 times). This demonstrates the power of using

transfer learning with transformers in dealing with texts. Especially with sentiment analysis field.

6. Work Environment and Development Tools

6.1. Hardware tools

To accomplish our work we used the following hardware configuration:

- Google Colab Hardware:
  - ✓ CPU : 2x Intel(R) Xeon(R) CPU @ 2.30GHz
  - ✓ RAM : 12 GB
  - ✓ Disk : 55 GB
  - ✓ GPU : Tesla T4
- Personal Hardware:
  - ✓ CPU : Intel(R) Core(TM) i5-6200U CPU@ 2.30Ghz
  - ✓ RAM : 8GB
  - ✓ Disk : 1T HDD
  - ✓ GPU : AMD Radeon (TM) R5 M335
  - ✓ Operating System : Microsoft Windows 8.1

6.2. Software tools

6.2.1. Google Colaboratory

Google Colaboratory, often shortened to "Colab" is a cloud service, offered by Google (free or paid), based on Jupyter Notebook and intended for training and research in machine learning. This platform makes it possible to train Machine Learning models directly in the cloud [50]. Without therefore needing to install anything on our computer except a browser. This environment allows us to write and execute Python code in your browser, with no configuration required, free access to GPUs, easy to share.

Figure 3.11- Google Colab logo

6.2.2. Python Programming language

Python is robust and user-friendly programming language created by Guido van Rossum and first Realized in 1991. It has simple and effective object-oriented programming techniques and high-level data structures. Python is an ideal language for scripting and rapid application development in many domains and on most platforms due to its easy syntax, dynamic typing, and the fact that it is interpreted [Web-13].

Python is one of the most popular programming language used by developers today, and it is by far the most used language in the field of Artificial Intelligence, most of the NLP and machine learning libraries are available in Python, and thus it is the best choice for our case.

Figure 3.12- Python logo

6.2.3. Used Libraries

- **PyTorch:** It is an open-source Machine learning framework built using python and the torch library. PyTorch uses tensors, which are multidimensional arrays designed for employing the GPU power for computational operations like matrix multiplication.

Figure 3.13- PyTorch logo

- **NumPy:** It is a library for Python programming language, intended to manipulate matrices or multidimensional arrays as well as mathematical functions operating on these arrays. More precisely, this free and open source software library provides multiple functions allowing in particular to directly create a table from a file or on the contrary to save a table in a file, and to manipulate vectors, matrices and polynomials. NumPy is the basis of SciPy, a grouping of Python libraries around scientific computing.

Figure 3.14- NumPy logo

- **Pandas:** Pandas is an open source Python library that provides powerful data manipulation and analysis tools. It is widely used in data science and data analysis workflows due to its efficient and flexible data structures. The primary data structure in Pandas is the DataFrame, which is a two-dimensional table-like data structure with labeled rows and columns. Pandas allows for easy loading, cleaning, transforming, and analyzing structured data, making it a popular choice for tasks such as data cleaning, data wrangling, exploratory data analysis, and data visualization. With its extensive range of functions and methods, Pandas simplifies the process of working with tabular data, enabling users to manipulate and extract insights from their data with ease.

Figure 3.15- Pandas logo

- **Transformers:** It is repository provided by Hugging Face which is a library built with python. It uses Pytorch and TensorFlow 2.0 frameworks to provides thousands of pretrained models to perform tasks on different modalities such as text, vision, and audio. As well as the orientation of that repository that tends to Natural language understanding (NLU) and Natural language generation (NLG) tasks which define the abstractive summarization. It contains a lot of features like its simplicity and entry-level code that makes researchers and educators execute and evaluate custom models faster, helper functions that allowed us to easily save checkpoints, and results while fine-tuning to make it easier for researchers to share their detailed results for more comparisons.

Figure 3.16- Transformers logo

The Table 3.13 provides additional information about the libraries we have used in our work:

Table 3.14- Versions of Python and the libraries used

Library	**Version**
Python	3.10.11
Torch	2.0.1+cu118
Numpy	1.22.4
Pandas	1.5.3
Transformers	4.29.2
Sklearn	1.2.2
Nltk	3.8.1
Matplotlib	3.7.1

7. Conclusion

In this chapter, we presented the different stages to creating the sentiment analysis model for the Algerian dialect based on transfer learning. Moreover, a series of experiments were carried out, which allowed us to fix the optimal values of the hyper-parameters, and specify the best pre-processing of our data. Then we present the evaluation of the model and the results obtained, as the model gave an accuracy of 81.88% on a test set, which are considered satisfactory and promising results. Finally, we showed the software and hardware tools used to develop our model.

General Conclusion

1. Conclusion

Sentiment analysis that has become a very popular area of research is essential in today's digitized world, especially in companies, and this is in order to improve their services and satisfy their customer. But there are still many problems facing this field, because sentiment analysis deals with unstructured text-based data on top of that this field is not very suitable for our society which has a very large dialectal variety.

In this context, we focused on the field of sentiment analysis of Feedback from the customers of Algerian telephone operators expressed in the Algerian dialect using deep learning. For this, we have proposed an approache, based on the fine-tuning of DziriBERT pre-trained model. An experimental study is performed, the purpose of which is to explore the hyper-parameters, and identify the pre-processing that provide the best results in terms of accuracy. From this study, we were able to define our model that gave an accuracy of 81.88 %. This is a very encouraging result.

2. Perspectives

As perspectives, we can cite:

- We intend to collect more data, and use Data Augmentation Techniques to reduce errors and increase model efficiency.
- In the near future, we plan to deploy our Model on a Web application. and use other Statistique techniques for sentiment anlysis of consumers

Bibliography

[1] Beysolow II, T. (2018). Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing. Apress.

[2] Hapke, H., Howard, C., & Lane, H. (2019). Natural Language Processing in Action: Understanding, analyzing, and generating text with Python. Simon and Schuster.

[**3]** Liu, B. (2012). Sentiment analysis and opinion mining [Online]. Morgan & Claypool Publishers.

[4] Jurafsky, D. (2018). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (1-135).

[5] Nasukawa, T., & Yi, J. (2003). Sentiment Analysis: Capturing Favorability Using Natural Language Processing. In Proceedings of the 2nd International Conference on Knowledge Capture, Florida, 23-25 October 2003 (pp. 70-77). doi: http://dx.doi.org/10.1145/945645.945658

[6] Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. The Web Conference.

[7] Taboada, M., Brooke, J., Tofiloski, M., et al. (2011). Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics [Online], 37(2), 267-307. Available on: https://www.mitpressjournals.org/doi/abs/10.1162/COLI_a_00049

[8] Liu, B. (2015). Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press.

[9] Potts, J. (2011). Creative industries and economic evolution. Edward Elgar.

[10] Pletea, D., Vasilescu, B., & Serebrenik, A. (2014). Security and Emotion: Sentiment Analysis of Security Discussions on GitHub. Proceedings of the 11th Working Conference on Mining Software Repositories [Online], 348–351.

[11] Hardeniya, T., & Borikar, D. A. (2016). Dictionary Based Approach to Sentiment Analysis - A Review. International Journal of Advanced Engineering, Management and Science (IJAEMS) [Online], 2(5), 317-322.

[12] Taboada, M. (2016). Sentiment analysis: An overview from linguistics. Annual Review of Linguistics, 2, 325-347. Doi: https://doi.org/10.1146/annurev-linguistics-011415-040518

[13] Zainuddin, N., Selamat, A., & Ibrahim, R. (2018). Hybrid sentiment classification on twitter aspect-based sentiment analysis. Applied Intelligence, 48, 1218-1232. Doi: https://doi.org/10.1007/s10489-017-1098-6

[14] Boudad, N., Faizi, R., Thami, R. O. H., & Chiheb, R. (2018). Sentiment analysis in Arabic: A review of the literature. Ain Shams Engineering Journal, 9(4), 2479-2490.

[15] Beesley, K. R. (2001, July). Finite-state morphological analysis and generation of Arabic at Xerox Research: Status and plans in 2001. In ACL Workshop on Arabic Language Processing: Status and Perspective (Vol. 1, pp. 1-8).

[16] Meftouh, K., Bouchemal, N., & Smaïli, K. (2012). A study of a non-resourced language: an Algerian dialect. In Spoken Language Technologies for Under-Resourced Languages.

[17] Sankoff, D. and Poplack, S. (1981). A formal grammar for code-switching. Research on Language & Social Interaction, 14(1):3–45. Doi: https://doi.org/10.1080/08351818109370523

[18] Pope, D., & Griffith, J. (2016, November). An Analysis of Online Twitter Sentiment Surrounding the European Refugee Crisis. In KDIR (pp. 299-306). Doi: https://doi.org/10.5220/0006051902990306

[19] Al-Ayyoub, M., Essa, S. B., & Alsmadi, I. (2015). Lexicon-based sentiment analysis of Arabic tweets. International Journal of Social Network Mining, 2(2), 101-114. Doi: https://doi.org/10.1504/IJSNM.2015.072280

[20] Al-Thubaity, A., Alqahtani, Q., & Aljandal, A. (2018). Sentiment lexicon for sentiment analysis of Saudi dialect tweets. Procedia computer science, 142, 301-307. Doi: https://doi.org/10.1016/j.procs.2018.10.494

[21] AL-Rubaiee, H. S., Qiu, R., Alomar, K., & Li, D. (2016). Sentiment analysis of Arabic tweets in e-learning. Journal of Computer Science, 12(11), 553-556.

[22] Elouardighi, A., Maghfour, M., Hammia, H., & Aazi, F. Z. (2018). Analyse des sentiments à partir des commentaires Facebook publiés en Arabe standard ou dialectal marocain par une approche d'apprentissage automatique. In EGC (pp. 329-334).

[23] Klouche, B., Benslimane, S. M., & Mahammed, N. (2021, January). Sentiment analysis of Algerian dialect using a deep learning approach. In International Conference on Artificial Intelligence and its Applications (pp. 122-131). Cham: Springer International Publishing. Doi: https://doi.org/10.1007/978-3-030-96311-8_12

[24] Sharma, S., & Bansal, M. (2018). Classification Approach for Sentiment Analysis. International Journal of Research in Electronics and Computer Engineering, 6(3), 1221-1224.

[25] Aldayel, H. K., & Azmi, A. M. (2016). Arabic tweets sentiment analysis–a hybrid scheme. Journal of Information Science, 42(6), 782-797. Doi: https://doi.org/10.1177/0165551515610513

[26] Emmert-Streib, F., Yli-Harja, O., & Dehmer, M. (2020). A clarification of misconceptions, myths and desired status of artificial intelligence. Doi: https://doi.org/10.48550/arXiv.2008.05607

[27] Ford, M. (2018). Architects of Intelligence: The truth about AI from the people building it. Packt Publishing Ltd.

[28] Asenjo Carvajal, A. (2020). Time to land prediction based on machine learning models (Bachelor's thesis, Universitat Politècnica de Catalunya). URL: http://hdl.handle.net/2117/329210

[29] Vargas, R., Mosavi, A., & Ruiz, R. (2018). Deep learning: a review. Doi: https://doi.org/10.20944/preprints201810.0218.v1

[30] Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow. Packt Publishing Ltd.

[31] Trahi, F. (2011). Prédiction de l'irradiation solaire globale pour la région de Tizi ouzou par les réseaux de neurones artificiels : Application pour le dimensionnement d'une installation photovoltaique pour l'alimentation du laboratoire de recherche LAMPA (Doctoral dissertation, Université Mouloud Mammeri). URL: https://www.ummto.dz/dspace/handle/ummto/623

[32] Kasri, S. (2010). Etude et modélisation de déclin de potentiel de surface par les réseaux de neurones (Doctoral dissertation, Université de Annaba-Badji Mokhtar).

[33] Feng, J., & Lu, S. (2019, June). Performance analysis of various activation functions in artificial neural networks. In *Journal of physics: conference series* (Vol. 1237, No. 2, p. 022030). IOP Publishing. Doi: https://dx.doi.org/10.1088/1742-6596/1237/2/022030

[34] Feng, J., He, X., Teng, Q., Ren, C., Chen, H., & Li, Y. (2019). Reconstruction of porous media from extremely limited information using conditional generative adversarial networks. *Physical Review E*, *100*(3), 033308. Doi: https://doi.org/10.1103/PhysRevE.100.033308

[35] Buduma, N. Locascio, N. (2017). Fundamentals of deep learning: Designing next-generation machine intelligence algorithms. -Sebastopol, CA, USA: O'Reilly Media.

[36] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551. Doi: https://psycnet.apa.org/doi/10.1162/neco.1989.1.4.541

[37] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. Neurocomputing, 187, 27-48. Doi: https://doi.org/10.1016/j.neucom.2015.09.116

[38] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011, June). Flexible, high performance convolutional neural networks for image classification. In Twenty-second international joint conference on artificial intelligence, 1237–1242. Doi: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-210

[39] Venugopal, P. Vigneswaran, T. (2019). State-of-Health estimation of li-ion batteries in electric vehicle using IndRNN under variable load condition. Energies, 12(22), 4338. Doi: https://doi.org/10.3390/en12224338

[40] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536. Doi: https://doi.org/10.1038/323533a0

[41] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780. Doi: https://doi.org/10.1162/neco.1997.9.8.1735

[42] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30, 6000-6010. Doi: https://dl.acm.org/doi/10.5555/3295222.3295349

[43] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. Doi: https://doi.org/10.48550/arXiv.1810.04805

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. Doi: https://doi.org/10.48550/arXiv.1910.10683

[45] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[46] El-Masri, M., Altrabsheh, N., Mansour, H., & Ramsay, A. (2017). A web-based tool for Arabic sentiment analysis. Procedia Computer Science, 117, 38-45. Doi: https://doi.org/10.1016/j.procs.2017.10.092

[47] Abdaoui, A., Berrimi, M., Oussalah, M., & Moussaoui, A. (2021). Dziribert: a pre-trained language model for the algerian dialect. Doi: https://doi.org/10.48550/arXiv.2109.12346

[48] Witten, I. H., Frank, E., Hall, M. A. (2011). Data Mining: Practical machine learning tools and techniques.3rd ed. Doi: https://doi.org/10.1016/C2009-0-19715-5

[49] Bouazizi, M., & Ohtsuki, T. O. (2016). A pattern-based approach for sarcasm detection on twitter. IEEE Access, 4, 5477-5488. Doi: https://doi.org/10.1109/ACCESS.2016.2594194

[50] Bisong, E. (2019). Building machine learning and deep learning models on Google cloud platform (pp. 59-64). Berkeley, CA: Apress.Doi: https://doi.org/10.1007/978-1-4842-4470-8

Webography

[Web-1] Ajay Khanna. (2022). Tokenization Techniques in Natural Language Processing in Python. URL: https://medium.com/@ajay_khanna/tokenization-techniques-in-natural-language-processing-67bb22088c75

[Web-2] Javier Ramos. (2021). Introduction to Natural Language Processing: NLP Tools For Python. URL: https://itnext.io/introduction-to-natural-language-processing-nlp-tools-for-python-cf39af3cfc64

[Web-3] Examples from geeksforgeeks of what short phrases look like with the stop words removed URL: https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

[Web-4] Vansh Jatana. ResearchGate. (2018). Dive into Machine Learning (PDF). URL: https://www.researchgate.net/publication/328614973_Dive_Into_Machine_Learning

[Web-5] Jonathan Buss. (2023). Activation function GELU in BERT. URL: https://iq.opengenus.org/activation-function-in-bert/

[Web-6] Pierre. (2020). Classification de pages Web via Deep Learning – Réseau de Neurones Convolutif. URL: https://www.anakeyn.com/2020/01/13/classification-de-pages-web-via-deep-learning-reseau-de-neurones-convolutif/

[Web-7] Afshine Amidi. (2019). CS 230 –Deep Learning. URL: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[Web-8] Andrew Kirillov. (2018). ANNT: Recurrent Neural Networks, URL: https://www.codeproject.com/Articles/1272354/ANNT-Recurrent-neural-networks

[Web-9] Lena voita. (2023). NLP Course-Seq2seq and Attention. URL: https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

[Web-10] Pantelis Monogioudis. (2023). Introduction to Transformers. URL: https://pantelis.github.io/data-mining/aiml-common/lectures/nlp/transformers/transformers-intro.html

[Web-11] DJOUGHI Mehdi. (2021). Algerian Dialect Review for sentiment analysis. URL: https://www.kaggle.com/datasets/djoughimehdi/algerian-dialect-review-for-sentiment-analysis

[Web-12] Charles Durfee. (2018). What is the Difference Between a Batch and an Epoch in a Neural Network?. URL: https://www.aiproblog.com/index.php/2018/07/19/what-is-the-difference-between-a-batch-and-an-epoch-in-a-neural-network/

[Web-13] "Le tutoriel Python — Documentation Python 3.11.4," 2023. URL : https://docs.python.org/fr/3/tutorial/