**Kasdi Merbah University of OUARGLA**

**Faculty of New Information and**

**Telecommunication Technologies**

**Department of:**

**Computer science and information technology**

# MASTER

Domain: Computer science

Field: Fundamental computing

Submitted by: **Bougoffa Asma Zahrat Arabea** and **Khediri Ahlem**

## Theme:

# Text Clustering in Social Media

**Evaluation Date: 18/06/2023**

**Before the Jury:**

| | | | |
|---|---|---|---|
| **Dr. Zerdoumi .O** | **MCA** | **President** | **UKM Ouargla** |
| **Dr. Mezati .M** | **MCB** | **Supervisor** | **UKM Ouargla** |
| **Dr. Mehjoub .B** | **MCB** | **Examiner** | **UKM Ouargla** |

## Acknowledgement

In the name of Allah, the most gracious, the most merciful. First, we are thankful to Almighty Allah for giving us the strength, knowledge, ability, and opportunity to realize this study and complete it satisfactorily.

Second, we would like to extend our sincere gratitude to Dr. MEZATI.M who served as our advisor, for his invaluable patience and guidance in helping us complete this thesis.

We would especially like to thank Mrs. SAADI.W for her advices throughout this work, and we would to thank the families khediri and bougoffa one by one.

## Abstract

With the increasing popularity of social media platforms like Twitter which is an important data source. Analyzing user-generated content become an important research area in the field of Natural Language Processing (NLP) and Machine Learning (ML). One of NLP and ML applications is text based emotion detection (TBED), which is a challenging task due to the informal nature of the text and the limited context provided specially the dialectical one. The main contribution of this work lies in the development and evaluation of an ensemble clustering approach for automatic labeling of text data according to the Ekman emotional model (happy, sad, angry, disgust, fear and surprise) from text in Algerian dialect derived from Twitter.

The proposed method combines multiple models of clustering algorithm to produce a single prediction. We utilized a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model specifically designed for multilingual text (including Arabic dialects), for the representation of the text in a dense vector space. The combination of ensemble clustering techniques with a multilingual BERT model shows promise in accurately capturing the nuanced emotions expressed in Algerian tweets. The findings of this research have implications for understanding public emotions, improving customer satisfaction analysis, and enhancing social media monitoring tools.



**Key words:** Machine Learning, Naturel Language Processing, Emotion Detection, Bert Multilingual, Ekman emotional model, Ensemble clustering.

## Résumé

Avec la popularité croissante des plateformes de médias sociaux comme Twitter qui est une source de données importante. L'analyse du contenu généré par l'utilisateur devient un domaine de recherche important dans le domaine du traitement du langage naturel (TLN) et de machine Learning (ML). L'une de leurs applications est la détection des émotions à partir du texte, qui est une tâche difficile en raison de la nature informelle du texte et du contexte limité fourni spécifiquement par la dialectique. La contribution principale de ce travail réside dans l'élaboration et l'évaluation d'une approche de regroupement d'ensemble pour l'annotation automatique des données textuelles selon le modèle émotionnel de Ekman à partir de textes a base du dialecte algérien dérivés de Twitter.

La méthode proposée combine plusieurs modèles d'algorithme de regroupement pour produire une seule prédiction. Nous avons utilisé un modèle BERT (Bidirectional Encoder Representations from Transformers) spécialement conçu pour le texte multilingue (y compris les dialectes arabes), pour la représentation du texte dans un espace vectoriel dense. La combinaison de techniques de regroupement d'ensemble avec un modèle BERT multilingue montre la promesse de capturer avec précision les émotions exprimées dans les Tweets algériens. Les résultats de cette recherche ont des implications pour la compréhension des émotions publiques, l'amélioration de l'analyse de la satisfaction des clients et l'accroissement des outils de surveillance des médias sociaux.



**Mots clés :** Machine Learning, Traitement du langage naturel, Détection des émotions, Bert multilingue, modèle émotionnel Ekman, Ensemble clustering.

## ملخص

مع تزايد شعبية منصات التواصل الاجتماعي مثل تويتر والتي تعتبر مصدر بيانات مهم. أصبح تحليل المحتوى الذي ينشئه المستخدم مجال بحث مهمًا في مجال معالجة اللغة الطبيعية (NLP) وتعلم الآلة (ML). أحد تطبيقات معالجة اللغة الطبيعية وتعلم الآلة هو اكتشاف المشاعر من النص، وهي مهمة صعبة بسبب الطبيعة غير الرسمية للنص والسياق المحدود المقدم خصيصًا باللهجات.

تكمن المساهمة الرئيسية لهذا العمل في تطوير وتقييم نهج التجميع الجماعي للتوسيم التلقائي للبيانات النصية وفقًا لنموذج Ekman العاطفي (سعيد، حزين، غاضب، متقزز، خوف ومفاجأة) من نص باللهجة الجزائرية مشتق من تويتر.

تجمع الطريقة المقترحة بين نماذج متعددة من خوارزمية التجميع لإنتاج تنبؤ واحد. استخدمنا نموذج BERT (تمثيلات التشفير ثنائية الاتجاه من المحولات) المُدرَّب مسبقًا والمصمم خصيصًا للنص متعدد اللغات (بما في ذلك اللهجات العربية)، لتمثيل النص في مساحة متجهة كثيفة. يُظهر الجمع بين تقنيات تجميع المجموعات مع نموذج BERT متعدد اللغات وعدًا في التقاط المشاعر الدقيقة المعبر عنها في التغريدات الجزائرية بدقة. نتائج هذا البحث لها آثار على فهم المشاعر العامة، وتحسين تحليل رضا العملاء، وتعزيز أدوات مراقبة وسائل التواصل الاجتماعي.



**الكلمات الأساسية:** التعلم الآلي، معالجة اللغة الطبيعية، اكتشاف المشاعر، Bert متعدد اللغات، نموذج Ekman العاطفي، تجميع المجموعات.

# Table of contents

**Chapter 3: Proposed solution and Implementation**



# Table of Figures

**Chapter 1: Emotion Detection in Social Media**



**Chapter 2 : Machine Learning**



**Chapter 3 : Proposed Solution and Implementation**

# List of Tables

# General Introduction

With the exponential growth of **social media** [23] platforms, an enormous amount of textual data is generated daily. Extracting valuable insights from this data requires sophisticated techniques based on **artificial intelligence** (AI) mainly **machine learning** (ML) techniques. By automatically analysing textual content, we can uncover underlying themes, trends, monitories and patterns with in the social media landscape.

The huge data created in social media platforms can be useful in many different and important aspects in high and accurate professional domains like comprehending the emotions and sentiments portrayed in social media text, this operation called emotion detection.

**Emotion detection** (ED) or emotion recognition is a branch of sentiment analysis that deals with the extraction and analysis of emotions. An emotion can be defined as psychological states differently connected with contemplations or as sentiments that result in physical changes reflect ones thoughts and conduct during given state.

The foundation for creating reliable models for **text clustering**, social media mining, and emotion recognition is provided by fusing machine learning with **natural language processing** (NLP). We can process and analyze textual data in a scalable and effective manner by utilizing algorithms and approaches from these domains.

In this report, we explore the methodologies, techniques, and challenges involved in applying text clustering in the context of emotion detection, and machine learning with NLP. We delve into algorithms such as **k-means** clustering, hierarchical clustering, and density-based clustering, while also considering advanced approaches like topic modelling and word embedding.

Furthermore, we explore the concept of **ensemble clustering** [47], delving into its methodology, benefits, and applications across various domains. Ensemble clustering involves creating an ensemble of individual clustering solutions and employing techniques such as clustering combination, consensus function selection, and ensemble member generation. By aggregating multiple clustering solutions, ensemble clustering provides a comprehensive perspective on the underlying data structure and enhances the reliability of clustering outcomes.

Our main goal is the automatic labelling of data (creating datasets automatically) in the context of Emotion detection. The resulted is a labelled data according to the basic **Ekman emotional model**.

The techniques used are based on the combination of clustering algorithms named ensemble clustering. In general, we can enhance the quality of clustering outputs, get deeper understanding of data patterns, and make better judgments based on the improved clustering solutions by using ensemble clustering. This process produces a labelled data that is prepared to be fed into supervised learning methods. In the context of emotion recognition for the **Arabic text**, especially the **dialect Algerian text**. [48]

The structure of this dissertation is started with chapter 1 where we will introduce different concepts related to the Emotion detection domain. In chapter 2, we will present different aspects related to the machine learning techniques precisely the clustering techniques. Finely the chapter 3, will describe our followed steps for creating automatically labelled dataset using clustering ensemble method in social media, we defend as well the used tools in the course of this works in addition to results and discussions.

# *Chapter 1*

# *Emotion Detection*

# *In social Media*

## 1.1. Introduction

Emotions are an integral part of human life and, among other things, highly influence decision-making. For that reason, emotion detection has become one of the most important aspects. Due to the almost endless applications of this new discipline, extracting them is a profitable venture for some companies. These companies have devoted a lot to developing various technologies based mainly on natural language processing and machine learning to extract emotions from different sources that enable the reading of emotions. Well, this technology can be applied in areas such as security, biometrics, law enforcement, etc…, for tracking and monitoring purposes.

## 1.2. Definition of Emotion

The Lexical definition of emotion is: "*A strong feeling deriving from one's circumstances, mood, or relationships with others.*" [1]

Emotions are responses to significant internal and external events. The term emotion is used to designate a collection of responses triggered from parts of the brain to the body, and from parts of the brain to other parts of the brain, using both neural and humoral routes. The result of the collection of such responses is an emotional state, defined by changes within the body proper. [2]

## 1.3. Emotions and Machines

A more practical approach, based on current technological capabilities, is the simulation of emotions in conversational agents in order to enrich and facilitate interactivity between human and machine. Affective computing is an emerging technology that has impacted various industries, from healthcare and education, to marketing and customer service, artificial intelligent emotion has brought many advantages to businesses operating in different areas. Although the use of affective artificial intelligence has remained a subject of controversy, specialists can agree that the innovative technology assists companies in better understanding their clients' needs and wants, which represents a valuable advantage in today's world.

## 1.4. Emotion Detection

Emotion detection **ED** is a branch of sentiment analysis that deals with extraction and analysis of emotions. Current affect detection systems are with respect to individual modalities or channels, such as face, voice and text. Emotion detection is the process of using artificial intelligence and machine learning techniques to identify, understand, and interpret human emotions from various sources such as text [3], speech, images, and videos, to detect emotions such as joy, sadness, anger, fear, and surprise. The application of emotion detection can be used in a variety of fields such as customer service, market research, mental health, and human-computer interaction.

## 1.5. Features of Emotion Detection Vs Sentiment Analysis

| | Emotion Detection | Sentiment Analysis |
|---|---|---|
| **Definition** | The process of identifying the emotions conveyed by a text, speech or image. | The process of identifying the overall sentiment or polarity of a text, speech or image. |
| **Objective** | To identify the specific emotions being expressed. | To identify the overall positive, negative, or neutral sentiment being expressed. |
| **Approach** | Relies on natural language processing and machine learning algorithms to identify specific emotions, such as joy, anger, sadness, or fear. | Relies on natural language processing and machine learning algorithms to analyse the overall sentiment or polarity of a text, speech, or image. |
| **Use Case** | Used in applications such as chatbots, voice assistants, and market research to detect the emotions of users or customers. | Used in applications such as social media monitoring, customer feedback analysis, and brand reputation management to understand the overall sentiment towards a product, brand, or topic. |
| **Limitations** | Emotions are subjective and can vary between individuals, cultures, and contexts. | Sentiment analysis may not always capture the nuances and complexities of human emotions and may miss sarcasm, irony, or other forms of figurative language. |
| **Example Output** | "This music makes me feel happy and energetic." (Emotion: Joy) | "I love this product! It's the best thing I've ever purchased." (Sentiment: Positive) |

**Table 1**: Emotion detection Vs sentiment analysis.

## 1.6. Emotion Models

Emotion models are the foundations of **ED** systems, they define how emotions are represented. The models assume that emotions exist in various states thus the need to distinguish between the various emotion states. When undertaking any **ED** related activity, it is imperative to initially define the model of emotion for use. Various forms of representing emotions are identified, however most importance is the categorical and dimensional emotion models.

### 1.6.1. Categorical Emotion Models

Categorical or **discrete emotion models (**DEMs) of emotions involves placing emotions into distinct classes or categories. Prominent among them include:

- **The Paul Ekman model** [4]categorizes emotions into **six fundamental categories**. According to the theory, there are six (06) essential emotions that arise from various brain systems as a result of how an experienced sees a circumstance, making emotions independent. These basic emotions are joy, sadness, anger, disgust, surprise, and fear.

- **The Robert Plutchik model** [5] presumes that there are a few core emotions that occur in opposite pairs and combine to generate complex emotions. In addition to the six basic emotions proposed by Ekman, he identified eight such fundamental emotions, including acceptance, trust, and anticipation. Joy vs. sadness, trust vs. disgust, anger vs. fear, and surprise vs. anticipation are the eight opposite emotions. According to Plutchik, there are variable degrees of intensity for each emotion as a function of how an experienced interprets events.

- **The Orthony, Clore,** and **Collins model** [6] disagreed with Ekman and Plutchik's comparison of "fundamental emotions." They did agree, however, that emotions emerged as a function of how individuals viewed events and that emotions varied in strength. They divided emotions into 22 categories, adding 16 to the basic emotions proposed by Ekman, resulting in a much broader representation of emotions, with additional classes of relief.

### 1.6.2. Dimensional Emotion Models

The **dimensional emotion model** (DiEMs) presupposes that emotions are not independent and that there exists a relation between them hence the need to place them in a spatial space. Thus, dimensional models position emotions on a dimensional space depicting how related emotions are and usually, reflecting the two main fundamental behavioural states of good and bad. Both unidimensional and multidimensional are affected by relative degrees (low to high) of their occurrences. Unidimensional models are rarely used but their fundamental idea permeates most multidimensional models.

- **Russell** introduces the **circumplex** of affect, a circular two-dimensional model [7]significant in dimensional emotion representation. The model distinguishes emotions in the Arousal and Valence domains, with Arousal distinguishing emotions by Activations and Deactivations and Valence distinguishing emotions by Pleasantness and Unpleasantness. The Circumplex model of Affect indicates that emotions are connected rather than autonomous. Ruseel's model is represented in **figure1**.


**Figure1**: Russell's circumplex model of affects.

- **Plutchik** presents a two-dimensional emotional wheel [4] with Valence on the vertical axis and Arousal on the horizontal axis. The emotions are depicted on the wheel in concentric circles, with the innermost emotions being derivatives of the eight fundamental emotions, followed by the eight fundamental emotions, and ultimately combinations of the primary emotions on the wheel's outermost regions. The wheel depicts how related emotions are based on their position on the wheel. Plutchik's emotional wheel is seen in **Figure2**.

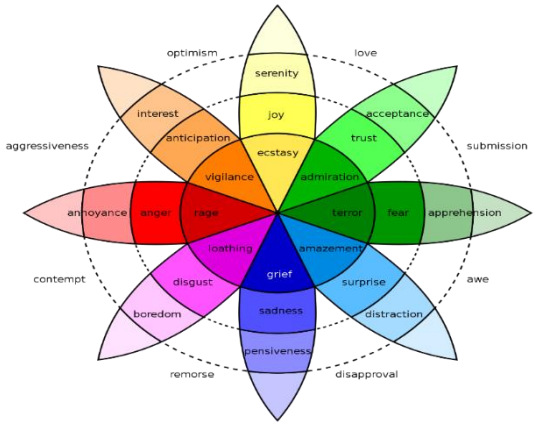



**Figure2**: Plutchik's wheel of emotions.

- **Russell** and **Mehrabian** also provide a three-dimensional emotion model [8] comprised of Valence/Pleasure, Arousal, and Dominance. According to the 2-D model, arousal and valence

describe how pleasant/unpleasant or active/inactive an emotion is. The third dimension of Dominance describes how much control experiencers had over their emotions.

## 1.7. Modalities for Emotion Detection

Emotions may be expressed by a person's speech, face expression and written text. Speech is a relevant communicational channel enriched with emotions: the voice in speech not only conveys a semantic message but also the information about the emotional state of the speaker. [9]

The human face (Facial expressions) is extremely expressive, able to express countless emotions without saying a word [10].Emotion detection from facial expressions refers to the process of identifying and categorizing an individual's emotions based on their facial expressions. This process involves using computer vision algorithms to analyse images or videos of a person's face and detect the subtle changes in facial features that are associated with different emotions.

Text Emotions can be shown in text-messages in two ways: With words and with orthography. Two potential problems associated with expressing emotions in text-messages are ambiguity of tone and disinhibited communicative behavior.

## 1.8. Text Based Emotion Detection (TBED)

Text data is a favorable research object for emotion detection [14] (recognition) when it is free and available everywhere in human life. Compare to other types of data, the storage of text data is lighter and easy to compress to the best performance due to the frequent repetition of words and characters in languages. Detecting emotions form text [12] has been an attractive task recent these years. Emotions can be extracted from two essential text forms: written texts and conversations (dialogues).

Emotion recognition in conversation extracts opinions between participants from massive conversational data in social platforms**,** such as Facebook**,** Twitter, YouTube, and others. ERC can take input data like text, audio, video or a combination form to detect several emotions such as fear, lust, pain, and pleasure.

## 1.9. Resources for Detecting Emotions in Text

Text Emotion detection is a relatively new field in natural language processing, and requires a large amount of annotated data for training and development. This section introduces some of the most prominent and publicly available sources, such as labelled texts and emotion lexicons, as well as vector space models.

### 1.9.1. Labelled Text

The Swiss Center for Affective Sciences (SCA) [13] provides ISEAR [14], a dataset of 7600 records of emotion provoking text. Balahur et al in [15] tackled the problem of emotion detection from another perspective, based on Appraisal Theory in psychology. They created a new knowledge base containing action chains and their corresponding emotion label. They clustered the examples within each emotion category based on language similarity and extracted the agent, the verb and the object from a selected subset of examples.

SemEval-2007 author in [16] and Alm's in [17] annotated fairy tale dataset has been used as benchmark in the literature. The lack of benchmark datasets with proper linguistic generality and accepted annotations pushes the research community to use text from microblogs, such as Twitter, to create an annotated corpus.

### 1.9.2. Emotion Lexicons

Having expressive emotional text like ISEAR is important for comparing different emotion detection [3] methods, but there are many use cases in which having an annotated lexicon could be useful. Authors in [18] created an emotion word lexicon, and [19] used Word Net-Affect to create a lexical representation of affective knowledge. Other works used crowd sourcing to annotate thirty-five thousand words. Other emotional lexicons are used in sentiment analysis, opinion mining, and emotion detection.

### 1.9.3. Word Embedding

Word embedding is a technique based on distributional semantic modelling, where words are represented as vectors in an n-dimensional space and the distance between vectors corresponds to the semantic similarity of the words they represent.

It has been shown to be useful in many natural language processing tasks, such as named entity recognition, machine translation, and parsing. Some of the more well- established and frequently used embedding models in the literature are latent semantic analysis and Word2Vec. There have been attempts to increase their performance, and incorporate more information in these models retrofitting and counter-fitting. Some work has been done in creating sentiment-specific word embedding using neural networks, to classify emotion in Twitter.

## 1.10. Approaches for Text based Emotion Detection

The different approaches proposed in the literature for the identification of emotions from the text were discussed in the following sections.

### 1.10.1. Keyword based approach

In this approach [20] the knowledge of key features is exploits, that are combination with **e**motion labels using a lexicon such as Word-Net Affect and SentiwordNet. Linguistic rules are applied and sentence structures are exploited. Further text pre-processing has to be performed on the given dataset, which includes stop words removal, tokenization and lemmatization. In addition, keyword spotting and **emotion** intensity are evaluated including with Negation checks. Finally, it determines the emotion label for each sentence.

### 1.10.2. Corpus based approach

Corpus-based emotion detection approaches [21]use supervised learning to induce sources of information such as word-emotion lexicons classified or weakly-labelled from a text corpus with a predefined collection of emotions extracted from emotion theories such as Ekman, Parrot, etc. More works are focused on lexicons, motivated by a considerable amount of study in the area of sentiment analysis.

### 1.10.3. Rule based Approach

To manipulate knowledge in order to view information in an advantageous way, the rule-based approach is used. It begins with text pre-processing initially, including stop word elimination**,** part - of-speech tagging, tokenization, etc. The rules of emotion are then derived using the concepts of statistics, linguistics, and computation. The best rules are selected later. Finally, the rules are applied to emotion datasets to determine the emotion labels. Subsequently, the appropriate rules are chosen.

### 1.10.4. Machine learning approach

In this case, emotion detection from text is based on classification problem involving different models from the disciplines of Natural Language Processing (NLP) and Machine Learning (ML). Machine learning is categorized into unsupervised learning and supervised learning. Naive Bayes (NB), Support vector machine (SVM), conditional random field etc., are the most common traditional unsupervised machine learning techniques used in this context. [22]

### 1.10.5. Hybrid Approach

In a unified model, the hybrid approach is a combination of different approaches. This approach has a higher likelihood of transcending the other approaches individually, leveraging the strength of the approaches used while trying to conceal their corresponding limitations. [20]

There is a several sources of text data amongst social media which led to the emergence of new opportunities and challenges for data analysis used for understanding user behavior.

Social media has become an important area of research because it represents a significant shift in the way that information is produced, circulated, and consumed in contemporary society.

## 1.11. Social Media

Social media [23] is an online platform and technology that allows individuals, organizations, and communities to create, share, and exchange information, ideas, and content. Social media platforms include popular sites such as Facebook, Twitter, Instagram, among others. It has become a major part of modern communication, allowing people to connect with each other, stay updated on news and events, and share their interests and perspectives with a wider audience.

### 1.11.1. Power of social media

Social media platforms such as Twitter and Facebook provide access to vast amounts of user-generated content, including text-based posts, comments, and messages, as well as images and videos. This data can be analyzed using clustering algorithms to group users or content based on common features or characteristics, such as shared interests, demographics, or behavioral patterns. The large amount of data available on social media makes it a valuable resource for clustering projects, as it provides a rich source of information that can be used to identify patterns and groupings among users or content.
By analyzing social media data, researchers can gain a better understanding of how people communicate, interact, and form communities in the digital age allowing businesses and researchers to monitor trends, sentiment and emotion as they happen. It also includes a variety of data types, such as text, images, videos, and audio, and user-generated data, which can provide a more accurate representation of public opinion and sentiment than traditional market research methods. The most popular platform used by persons are Facebook, Twitter…etc.

### 1.11.2. Twitter

Twitter [24] is a widely used free social networking tool that allows people to share information, where users broadcast short posts known as tweets. These tweets can contain text, videos, photos or links. It is a microblogging service for registered users to post, share, like and reply to tweets with short messages. Nonregistered users can only read tweets.

Every day Twitter provide a huge among of data, which can be effectively utilized in the areas of academic research, social projects, and studying marketing methodologies such as: Topic Modeling, Network Analysis, Sentiment Analysis and emotion detection.

We as humans can discern emotional sentiments in writings and conversations, but computer systems can grasp the emotion of a document as well as its literal meaning with the assistance of natural language processing.

## 1.12. Naturel Language Processing NLP

Natural language processing (NLP) [25] is a subset of computer science closely associated with machine learning that focuses on allowing machines to understand the natural language used for communication among humans. Some examples of tasks performed using NLP include sentiment analysis, speech recognition, and the automatic generation of responses to questions. Other famous NLP applications include voice assistants like Alexa [26], Google Home [27], or even Siri [28]

NLP has many applications, spreading its wings in almost every field. Help decrease manual labour and do the tasks accurately and efficiently. We can use NLP in email filtering, language translation, document analysis, sentiment analysis and Emotion Detection. The table below illustrate examples for some applications of NLP:

| Applications | Examples |
|---|---|
| Sentiments analysis & Emotion detection | Community morale monitoring \| Product review triangle \| Costumer care |
| Search | Documents \| Web |
| Editing | Style \| Grammar \| Spelling |
| Dialog | Virtual assistants \| Scheduling \| Chatbots |
| Writing | Index \| Concordance \| Table of contents |
| Email | Spam filter \| Classification\| Prioritization |
| Text mining | Summarization \| Medical diagnoses \| Knowledge extraction |
| Attribution | Plagiarism detector\| Literary forensics \| Style coaching |
| Behaviour detection | Finance \| Election forecasting \| Marketing |
| News | Event detection \| Fact checking \| Headline composition |
| Creative writing | Movie scripts \| Poetry \| Song lyrics |

**Table 2**: Different applications of NLP.

## 1.13. Conclusion

In this chapter, we introduced emotion detection (**ED**), especially from textual data. We started by giving the definition of the concepts related to emotion detection in all its aspects and we took an overview of the difference between emotion detection and sentiment analyses and their models also modalities used for them. Then we further talked about text Based Emotion Detection (**TBED**) and its resources, finally we presented approaches related to the domain including natural language processing and their applications and some examples of them.

In the next chapter, we will outline our strategy for clustering from social media beginning by short presentation of related key concepts.

# *Chapter 2*

# *Machine Learning*

## 2.1. Introduction

Artificial Intelligence (**AI)** is a rapidly advancing field that focuses on creating intelligent systems capable of performing tasks that typically require human intelligence. Machine Learning (**ML)** is a subset of AI that involves the development of algorithms and models that enable computers to learn from data and make predictions or decisions without being explicitly programmed. **ML** is a vast science allows the development of several areas.

## 2.2. Machine learning definition

Machine learning was defined in 90's by *Arthur Samuel* described as the, *"it is a field of study that gives the ability to the computer for self-learn without being explicitly programmed*" [29]**,** that means imbuing knowledge to machines without hard coding it.

Another definition: "*A computer algorithm/program is said to learn from performance measure P and experience E with some class of tasks T if its performance at tasks in T, as measured by P, improves with experience E*". [30]

Machine learning is mainly focusing on the development of computer programs, which can teach themselves to grow and change when exposed to new data. Machine learning studies algorithms for self-learning to do stuff. It can process massive data faster with the learning algorithm. For instance, it will be interested in learning to complete a task, make accurate predictions, or behave intelligently.

## 2.3. Types of machine Learning

Machine Learning is a vast field that includes various approaches and techniques, it is mainly divided into four categories, which are as follows:

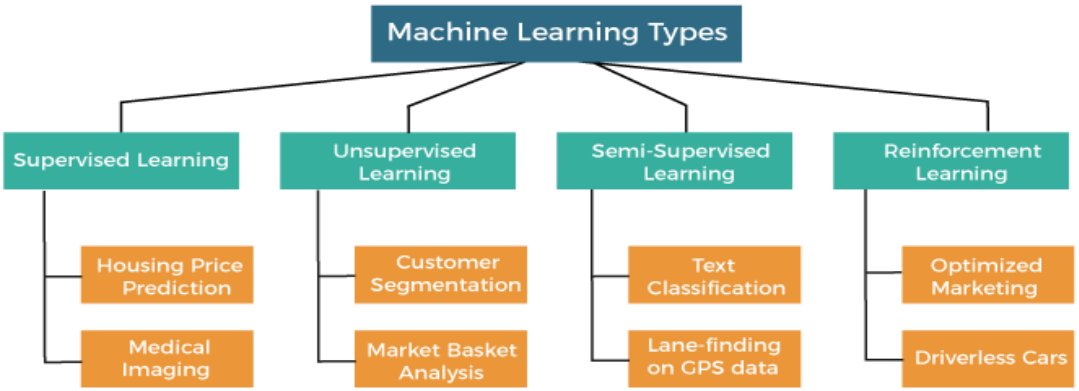



**Figure 3**: Machine Learning Types.

### 2.3.1. Supervised Learning

Supervised Learning is the first type of machine learning, in which marked data (labelled data) is used to train the algorithms, where the input and the output are known.

Supervised learning uses the data patterns to predict the values of additional data for the labels. This method will commonly use in applications where historical data predict likely upcoming events. [31].

The Supervised Learning mainly divided into two parts, which are:

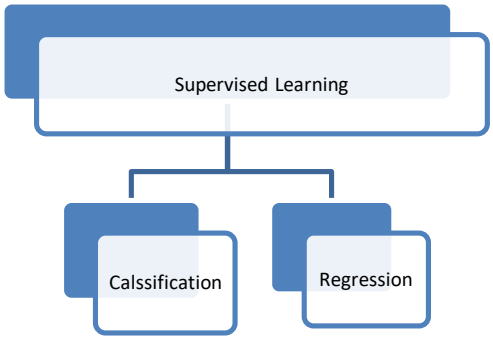



**Figure 4**: Supervised Learning Types.

**2.3.2.1. Regression:** is the type of Supervised Learning in which labelled data used, and this data is used to make predictions in a continuous form. The output is always continuing, and the graph is linear. Regression [32] is a form of predictive modelling technique, which investigates the relationship between a dependent variable ***Outputs*** and independent variable ***Inputs***. This technique used for forecasting the weather, time series modelling and process optimisation.
There are many Regression algorithms present in machine learning, which are used for different regression applications. Some of the main regression algorithms are illustrated in [32]**.**

**2.3.2.2. Classification:** is the type of Supervised Learning in which labelled data is used to make predictions in a non-continuous form. The output of the information is not always continuous, and the graph is non-linear. In the classification technique [33], the algorithm learns from the data input given to it and then uses this learning to classify new observation. This data set may merely be bi-class, or it may be multi-class too. [33] One of the examples of classification problems is to check whether the email is spam or not spam by train the algorithm for different spam words or emails.

### 2.3.2. Semi-Supervised learning

Semi-supervised learning [34] is a type of machine learning. That uses a small portion of labelled data and many unlabelled data to train a predictive model. It refers to a learning problem (and algorithms designed for the learning problem) that involves a small portion of labelled examples and a large number of unlabelled examples from which a model must learn and make predictions on new examples. This type sits between supervised learning and unsupervised learning.

### 2.3.3. Deep Learning

Deep learning [35] is essentially a neural network with multiple layers. It is a sub-set of Machine Learning which learns multiple levels of representation of data. In deep the higher-level features are learned from the lower-level features. A deep learning model can be trained to recognize patterns and make predictions in a variety of applications, such as image recognition, natural language processing, and speech recognition.

### 2.3.4. Reinforcement Learning

Reinforcement Learning is a type of machine learning in which no raw data is given as input, its algorithms [36] have to figures out the situation on their own. With reinforcement learning, the algorithm discovers through trial and error which actions yield the most significant rewards. This type of training has three main components, which are the agent, which can describe as the learner or decision maker, the environment, which described as everything the agent interacts with, and actions, which represented as what the agent can do. The reinforcement learning frequently used for robotics, gaming, and navigation.

### 2.3.5. Unsupervised Learning

Unsupervised Learning is the second type of machine learning, in which unlabelled data are used to train the algorithm, which means it used against data that has no historical labels. The purpose is to explore the data and find some structure within. In unsupervised learning [37], the data is unlabelled and the input of raw information directly to the algorithm and without knowing the output of it. The data cannot divide into a train or test data. The algorithm figures out the data and according to the data segments, it makes clusters of data with new labels.

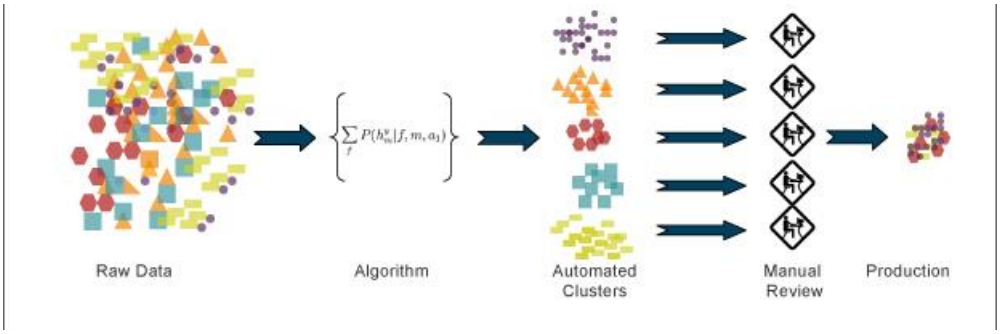



**Figure 5**: Unsupervised Learning.

The Unsupervised Learning mainly divided into two parts, which are presented in **figure 6**.

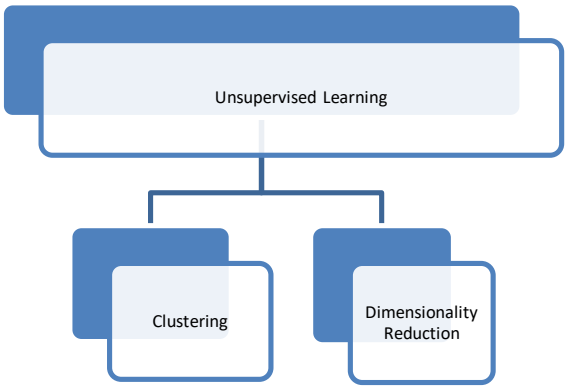



**Figure 6**: Unsupervised Learning Types.

**2.3.2.1. Dimensionality Reduction:** is the type of Unsupervised Learning, in which the dimensions of the data is reduced to remove the unwanted data from the input. This technique is used to remove the undesirable features of the data. It relates to the process of converting a set of data having large dimensions into data with carries same data and small sizes. These techniques used while solving machine learning problems to obtain better features.

There are many Dimensionality reduction [34] algorithms present in machine learning, which applied for different dimensionality reduction applications.

**2.3.2.2. Clustering:** is the type of Unsupervised Learning [38] in which unlabelled data is used, it is the process of grouping similar entities together, and then the grouped data used to make clusters. A cluster is a group of data points that are similar to each other based on their relation to surrounding data points. Clustering used for things like feature engineering or pattern discovery. The goal of this unsupervised machine learning technique is to find similarities in the data point and group similar data points together and to figures out that new data should belong to which cluster.

There are different types of clustering algorithms that handle all kinds of unique data.

**1/ Density-based**: In density-based clustering [38] [39], data grouped by areas of high concentrations of data points surrounded by areas of low concentrations of data points. The algorithm finds the places that are dense with data points and calls those clusters. The clusters can be any shape, which is great thing; we are not constrained to expected conditions. The clustering algorithms under this type do not try to assign outliers to clusters, so they get ignored.

**2/ Distribution-based:** with a distribution-based clustering [40] approach, all of the data points are considered parts of a cluster based on the probability that they belong to a given cluster.
It works like this: there is a center-point, and as the distance of a data point from the center increases, the probability of it being a part of that cluster decreases.

**3/ Centroid-based:** [40] is a little sensitive to the initial parameters given to it, but it's fast and efficient. These types of algorithms separate data points based on multiple centroids in the data. Each data point is assigned to a cluster based on its squared distance from the centroid. This is the most commonly used type of clustering.

**4/ Hierarchical-based:** [40] is typically used on hierarchical data, as we would get from a company database or taxonomies. It builds a tree of clusters so everything is organized from the top-down. This is more restrictive than the other clustering types, but it is perfect for specific kinds of data sets.

**2.3.2.3. Clustering Algorithms**

**1/ K-Means Clustering:** [38] is an iterative clustering algorithm that aims to find local maxima in each iteration. It starts with K as the input, which is how many groups you want to see. Input k centroids in random locations in your space. The Euclidean distance method is used to calculate the distance between data points and centroids, and assign data point to the cluster, which is close to it. Recalculate the cluster centres as a mean of data points attached to it. Repeat until no further changes occur.

**2/ Hierarchical Clustering:** is one of the algorithms of Clustering technique, in which similar data grouped in a cluster. An algorithm builds the hierarchy of clusters. This algorithm starts with all the data points assigned to a bunch of their own. Then two nearest groups are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

**3/ DBSCAN clustering:** stands for density-based spatial clustering of applications with noise. It is a density-based clustering algorithm. This is an appropriate algorithm for finding outliners in a data set. It finds arbitrarily shaped clusters based on the density of data points in different regions. It separates regions by areas of low-density so that it can detect outliers between the high-density clusters.

**4/ Gaussian Mixture Model Clustering:** [41] uses multiple Gaussian distributions to fit arbitrarily shaped data. In general, data needs to follow a circular format. The way other algorithms calculate the distance between data points has to do with a circular path, so non-circular data is not clustered correctly. Gaussian mixture models fix this issue, there is no need circular shaped data for it to work well.

**5/ FUZZY C-Means Clustering:** [42] is a data clustering technique where each data point belongs to a cluster to a degree that is specified by a membership grade. It starts with an initial guess for the cluster centers, which represent the mean location of each cluster.

**6/ Agglomerative Hierarchy clustering:** is a common type of hierarchical clustering [43] algorithm used to group objects in clusters based on how similar they are to each other. This is a form of bottom-up clustering, where each data point is assigned to its own cluster. Then those clusters get joined together. At each iteration, similar clusters are merged until all of the data points are part of one big root cluster.

In the table below, we will illustrate a comparative study of some most used clustering algorithms.

| Clustering Algorithm | Positive points | Negative points |
|---|---|---|
| **K-means** | It scales to large data sets. simple to implement. guarantees convergence. can warm-start the positions of centroids. It easily adapts to new examples. Generalizes to clusters of different shapes and sizes. | K-means has trouble clustering data where clusters are of varying sizes and density. |
| **DBSCAN** | Does not require one to specify the number of clusters in the data a priori, as opposed to k-means. Can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster. | Does not work to well when we are dealing with clusters of varying densities or with high dimensional data. |
| **Agglomerative** | It works from the dissimilarities between the objects to be grouped together. A type of dissimilarity can be suited to the subject studied and the nature of the data. The dendrogram output of the algorithm can be used to understand the big picture as well as the groups in your data. | The time and space complexity of agglomerative clustering is more than K-means clustering, and in some cases, it is prohibitive. |

| | | |
|---|---|---|
| **Fuzzy Clustering** | it allows gradual memberships of data points to clusters measured as degrees in [0,1]. This gives the flexibility to express that data points can belong to more than one cluster. | is the randomness of the initial clustering center, which usually leads to the local optimal solutions and have a great influence on the clustering results. |

**Table 3**: Comparison of previous clustering algorithms.

Each of the mentioned clustering algorithms are designed on a particular assumption, which is normally realized via input parameters. Generally, there is no clustering algorithm that performs well for every set of data. To overcome the sited problems and to identify a proper alternative and improve the clustering results, the methodology of clustering ensemble has been developed in the past decade.

## 2.4. Ensemble Clustering

The combination of groups of classifiers in order to improve accuracy is a well-established strategy in supervised learning. This technique can also be applied to unsupervised learning. The assumption is that combining different partitions it is possible to obtain a consensuated partition that improves over the individual ones, thus the name of consensus clustering [44], also known as clustering ensemble [45].

Clustering ensemble [38] technique involves running multiple clustering algorithms on the same dataset and combining their results to obtain a more accurate clustering. Which produce a better result than that of the individual clustering algorithms in terms of consistency and quality.

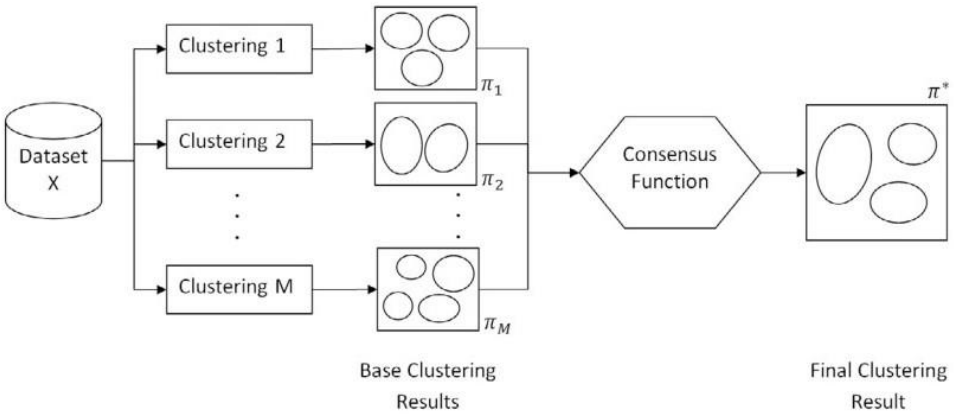



**Figure 7:** Ensemble Clustering.

The goal of consensus clustering is to obtain a new partition that uses the information of all n partitions with robustness, consistency, stability, and novelty.

### 2.4.1. Ensemble clustering strength

Consensus clustering adds advantages over classical clustering algorithms, such as: [38]

- Knowledge reuse, given that the consensus can be computed directly from the partition assignments, previous partitions from the data using the same or different attributes can be introduced in the process.
- Distributed computing, the individual partitions can be obtained independently, so the computational cost of obtaining the partitions to consensuate can be distributed in different processes.
- Privacy, only the assignments of the individual partitions are needed for the consensus, so partitions that use attributes with sensitive information do not need to be shared to obtain the final partition.

### 2.4.2. Ensemble clustering process

Clustering ensemble is a two main step process of generating individual partitions and combining them to generate the final partition.

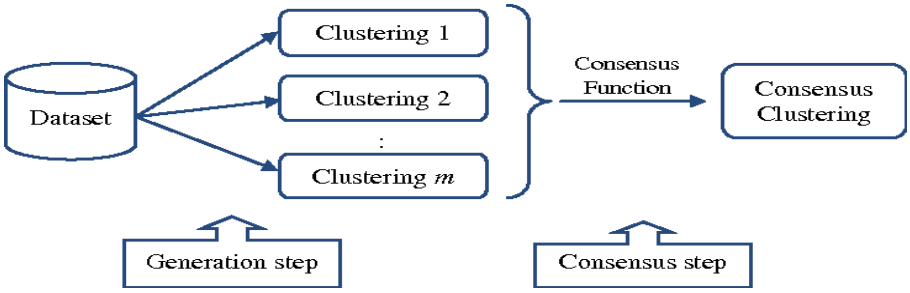



**Figure 8**: Diagram of the general process of ensemble clustering [47].

### 2.4.2.1. Generation Step

Partition generation is the process [38] of generating multiple partitions of a data set using different clustering algorithms.

One of the main ideas of ensemble generation is that the key to a better performance resides in the quality and diversity of the individual classifiers to combine.

In the generation step there are no constraints about how the partitions must be obtained. Therefore, in the generation process different clustering algorithms [47] or the same algorithm with different parameters initialization can be applied. Even different objects representations, different subsets of objects or projections of the objects on different subspaces could be used.

There are different techniques used for generating the individual clusters: [38]

**1/** Different example representations: Diversity is achieved by generating partitions using different subsets of data attributes, allowing for partitions that use different and complementary perspectives of the data.

**2/** Different clustering algorithms: to take advantage that all clustering algorithms have different biases and that can produce different partitions using the same data.

**3/** Different parameter initialization: some clustering algorithms are able to produce different partitions using different parameters. This includes parameters that change directly or indirectly the number of clusters and the starting point of the algorithms based on an objective function that is optimized using local search (for instance k-means).

**4/** Subspace projection: Using dimensionality reductions techniques like random projections or random cuts allow producing different clusterings from different perspectives.

**5/** Subsets of examples: Based also on techniques used by supervised ensembles, random subsamples (bootstrapping) of the dataset allow generating diverse clustering.

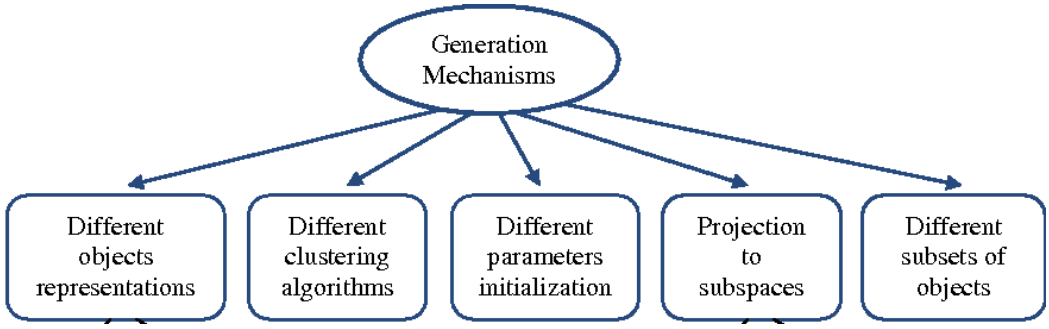



**Figure 9**: Principal ensemble clustering generation mechanisms. [47]

#### 2.4.2.2. Consensus process

The consensus process is the process of combining multiple clustering of the same data set to produce a single more robust clustering. The consensus process uses the information of the assignments of the examples to determine the consensuated partition. Techniques can be divided into two groups: those based on the objects co-ocurrence matrix and those based on the median partition problem. [47]

**1/ Co-ocurrence based methods**: This group of methods uses the co-occurrence of objects in the different clustering to determine the consensus clustering. The idea is that if two objects are frequently co-occurred in the different clustering, then they are likely to belong to the same cluster in the consensus clustering.

These methods use the labels obtained from each individual clustering and the coincidence of the labels for the examples in different clustering. There are several ways to exploit this information, the main methods are based on relabeling and voting, the co-association matrix, graph and hyper-graph partitioning, information theory measures and finite mixture models.

**2/ Median partition-based methods**: This group of methods assumes that there is a partition that is equally similar to all the individual clustering. This partition is then used as the consensus clustering. There are several similarity functions that can be used to compute the similarity among partitions, most of them are based on external cluster validity indices.

## 2.5. Conclusion

Machine learning is a dynamic and rapidly evolving field that has revolutionized industries all over the world. We tried in this chapter to explain different concepts related to clustering. Introduced by, describing some key concepts related to the machine learning domain. Then, we presented some algorithms from different category's precisely K-Means Clustering in unsupervised. Finally, we presented clustering ensemble mechanism and their different techniques.

In the next chapter, we will present the description of the design and the implementation of our work in the context of Emotion detection using the clustering ensemble techniques.

# *Chapter 3*

# *Proposed solution and Implementation*

## 3.1. Introduction

Twitter is considered one of the most important social media applications that have a huge data, since most data analysts and researchers turn to it because of the short text characteristic of tweets, especially who is interested in emotion detection and sentiment analysis. Our work is in the context extracting emotions from dialectical Algerian tweets by combining similar tweets using clustering techniques.

Our choice of the Algerian dialect represents a big challenge because of the difficulties related to the dialect that has words with different languages, which make the preprocessing step a difficult part. Well, that explains the small number of works that are based on it, comparing with others like English works. The used clustering techniques, we focused in our work on ensemble clustering to group tweets having similar emotion.

In this chapter, we will try to explain different steps of our contribution in the context of emotion detection using clustering ensemble techniques.

## 3.2. Motivation

The two main problems encountering researchers in the field of clustering, especially texts clustering, are collecting dataset and choosing the suitable clustering technique for this dataset. Where the two of them represent a difficult challenge. For the collecting data problem, social media represent the most powerful source of this because of the huge amount of information on its platforms like Twitter. However, we are facing the obsessive use of the dialectical languages, which creates problems in data handling.

The Algerian dialect refers to the dialect spoken in Algeria. Its words are mostly derived from the Arabic language, though it has also borrowed many words from French, and, to lesser extent, from Berber, Turkish, and rare words derived from Spanish. [48] In the context of emotion detection from text, there are several studies on English and other languages; however, there are fewer researches on Arabic language especially dialectical one.

Frequently used techniques for text handling problems, are those based on machine learning. There is a crucial need of dataset (labeled data) to perform classification tasks based on ML techniques; however, there is less Arabic datasets, especially those destined for Algerian Dialect.

Generally, the labeling of data is manually realized, but clustering techniques is the alternative solution for automatic data labeling.

The choice of clustering technique is the main factor affecting the clustering results. When applying clustering algorithms to a set of objects, it imposes an organization to the data following an internal

criterion and the characteristics of the used (dis)similarity function. Different solutions obtained by different clustering algorithms can be equally plausible, if there is no previous knowledge about the best way to evaluate the results. The idea of combining different clustering results (cluster ensemble or clustering aggregation) emerged as an alternative approach for improving the quality of the results of clustering algorithms.

Even if we select the technique, we still must choose the suitable parameters of this technique to get the best result possible, for this problem clustering ensemble was provided, where the problem hand over to be addressed by this technique.

## 3.3. Contribution

To overcome the previous cited difficulties related to the emotion detection from dialectical Algerian text, we give an alternative solution-based on clustering ensemble techniques to automatically label and create an Algerian dataset based on twitter data that can be used in emotions detection tasks.

## 3.4. Related works

The short form of twitter text is the main attractive characteristic making an important number of studies emerging in the last years. Twitter data sets can be effectively utilized in the areas of academic research, social projects, and studying marketing methodologies using different techniques for several purpose. We will mention bellow some of works based on Twitter text.

**1/ Sentiment Analysis:** for sentiment analysis goals, [49] involves classifying tweets as positive, negative, or neutral. Using SVM classifier combined with a cluster ensemble can offer better classification accuracies than a stand-alone SVM. Where they employed an algorithm, named C3E-SL, capable to combine classifier and cluster ensembles. Which can refine tweet classifications from additional information provided by clusters, assuming that similar instances from the same clusters are more likely to share the same class label.

**2/ Emotion detection**: in [21] authors analyzed how emotions are distributed in the data they annotated and compared it to the distributions in other emotion-annotated corpora, and they used the annotated corpus to train a classifier that automatically discovers the emotions in tweets. In addition, they presented an analysis of the linguistic style used for expressing emotions.

**3/ Topic Modeling:** Topic modeling involves identifying and analyzing the themes and topics discussed in tweets. In [50] Extended Twitter-LDA in two ways. Modeling the generation process of tweets more accurately by estimating the ratio between topic words and general words for each user. And enable it

to estimate the dynamics of user interests and topic trends online based on the Topic Tracking Model (TTM), which models consumer purchase behaviors.

**4/ Network Analysis:** Network analysis techniques, such as centrality analysis and community detection, can be used to identify influential users, study the structure of social networks, and track the spread of information through re tweets and mentions. In [51] authors applied SNA as a theoretical and methodological framework to demonstrate that the interactions among users are different when a whole network is analyzed, and when it is divided into the mentions and retweets networks. By doing this, hidden patterns are revealed.

**5/ Crisis Management:** Twitter data has also been used for crisis management, such as predicting and monitoring natural disasters or tracking the spread of disease outbreaks. The work [52] considered how emergency response organizations utilize available social media technologies to communicate with the public in emergencies and to potentially collect valuable information using the public as sources of information on the ground. The authors discuss the use of public social media tools from the emergency management professional's viewpoint with a particular focus on the use of Twitter. Limited research has investigated Twitter.

**6/ Political Analysis:** Twitter data has been widely used to study political discourse and sentiment. [53] Authors addressed a methodology to predict the outcome of the 2019 Indian general elections using the sentiment analysis of twitter data. Where a decision tree classifier is used to train and test data and the predicted outcome is found to be close to that of the actual outcome and most of the pre poll analysis done so far.

The classification problems need labeled data, the process of labeling data is mostly realized manually, therefore there is many works in the automatic data labeling. The next section highlights the stat of the art using clustering ensemble techniques for numerous objectives precisely automatic data labeling.

**1/ Medical diagnosis:** The process of identifying a disease, condition, or injury from its signs and symptoms. A health history, physical exam, and tests, such as blood tests, imaging tests, and biopsies, may be used to help make a diagnosis. This paper [44] present a new methodology of class discovery and clustering validation tailored to the task of analyzing gene expression data. The method can best be thought of as an analysis approach, to guide and assist in the use of any of a wide range of available clustering algorithms.

**2/ Fraud detection:** Ensemble clustering can be used to detect fraudulent transactions by clustering transactions with similar characteristics. This can help banks and other financial institutions to prevent fraud. In the article [54] authors proposed a heterogeneous ensemble learning model based on data distribution (HELMDD) to deal with imbalanced data in CCFD. In addition, they validate the effectiveness of HELMDD on two real credit card datasets.

**3/ Text mining**: Ensemble clustering can be used to cluster documents into different topics. This can help to improve the accuracy of text classification and to identify the most important information in a document. In this work [55], the authors presented a two-stage framework for topic extraction from scientific literature. It employs a two-staged procedure, where word embedding schemes are utilized in conjunction with cluster analysis. An improved word embedding scheme is proposed, which incorporates word vectors obtained by word2vec, POS2vec, word-position2vec and LDA2vec schemes. In the clustering phase, an improved clustering ensemble framework is presented, which incorporates conventional clustering methods such as k-means, k-modes, k-means++, self-organizing maps and DIANA algorithm. The empirical analysis reveals that ensemble word embedding scheme yields better predictive performance compared to the baseline word vectors for topic extraction.

## 3.5. Conception

The process of the dataset creation is divided into many steps, the next section is reserved to explain our steps details. The diagram in **Figure 10** represents the different steps of our system.

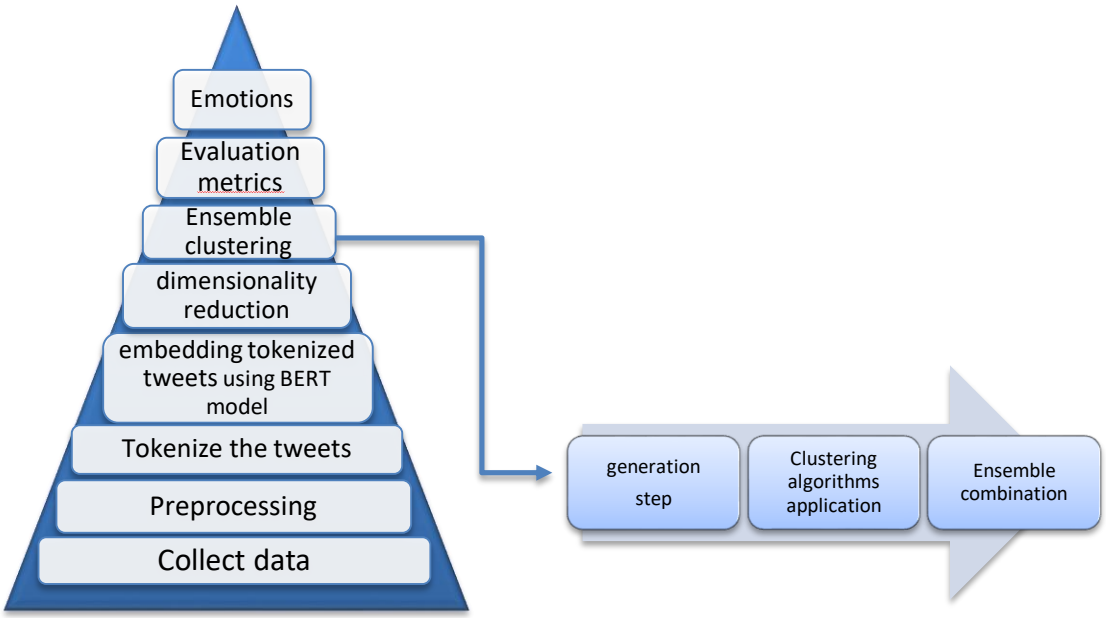



**Figure 10:** Realization steps of Emotion Detection.

### 3.5.1. Collect Data

There are many sources provide databases, in our data collection we used two famous sources which are Twitter API and Kaggel.

**1/ Twitter API:** Twitter provides access to its public data through its API (Application Programming Interface), which is a set of programmatic endpoints that can be used to understand or build the conversation on Twitter. Twitter API allows developers to retrieve tweets and other information from the platform like Tweets, Users, Spaces, Direct Messages, Lists, Trends, Media, and Places.

**2/ Kaggel:** Kaggel [56] is the world's largest data science community with powerful tools and resources to help you achieve your data science goals. Kaggle offers a no-setup, customizable, Jupyter Notebooks environment. Access GPUs at no cost and a huge repository of community published data and code.

We used the official Twitter API to retrieve data, and returns some information, which includes the following: user (string): Username. Text (string): Tweet (280 characters). Created_at (Date): The time

when the post was created. In addition, we used a data from the platform kaggel; an unlabelled Algerian data [57] which has 4134 tweets.

### 3.5.2. Data Pre-processing

For cleaning the data, we did the following steps:

**1/ Text Cleaning:** Text cleaning helps ensure consistency and improves the quality of the input data. Depending on our task, this involve removing special characters like removing punctuation, numbers, URLs, remove mentions, stock market tickers like $GE, old style retweet text. In addition, for the Arabic words cleaning, we removed diacritics, Letter lengthening, Arabic question mark.

**2/ Hashtags:** A hashtag is a combination of letters, numbers, and/or emoji preceded by the # symbol. Hashtags are used to categorize or label content and make it more discoverable. Therefore, we find that it contains an information that can be used in our task. Therefore, we just removed the hash # sign and kept the words with replacing the separator symbol with a space to turn it into a regular sentence.

**3/ Lowercasing:** Converts text to lowercase. This step helps to standardize the text and prevent the model from treating the same word in different cases as different symbols. We know that some Algerian tweets contain English or French words, which are the ones concerned with this step.

**4/ Emoji Handling:** The term "emoji" refers to any little images, icons, or symbols that are used in text messages, social media, email, or any other text field in electronic communication. Users utilize them to describe their emotional state, provide information fast and clearly, send playful messages without using words.

In emotion detection task removing the emoji is not the right decision, therefore we handle emojis with more than a way to find the best result in the detection of the emotion residing in the text:

- Replace emojis with their description using a python package (**emoji**):

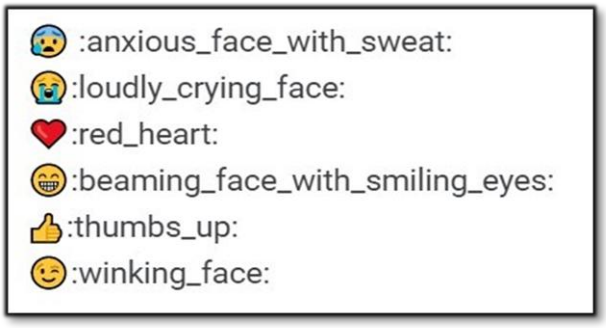



**Figure 11:** Emojis description.

- We replace a list of emojis that express the same emotion with one emoji by creating an emoji map, and finely replacing the resulting emoji with his meaning.(the list of emojis was created based on the most popular emojis used in those last year [58])

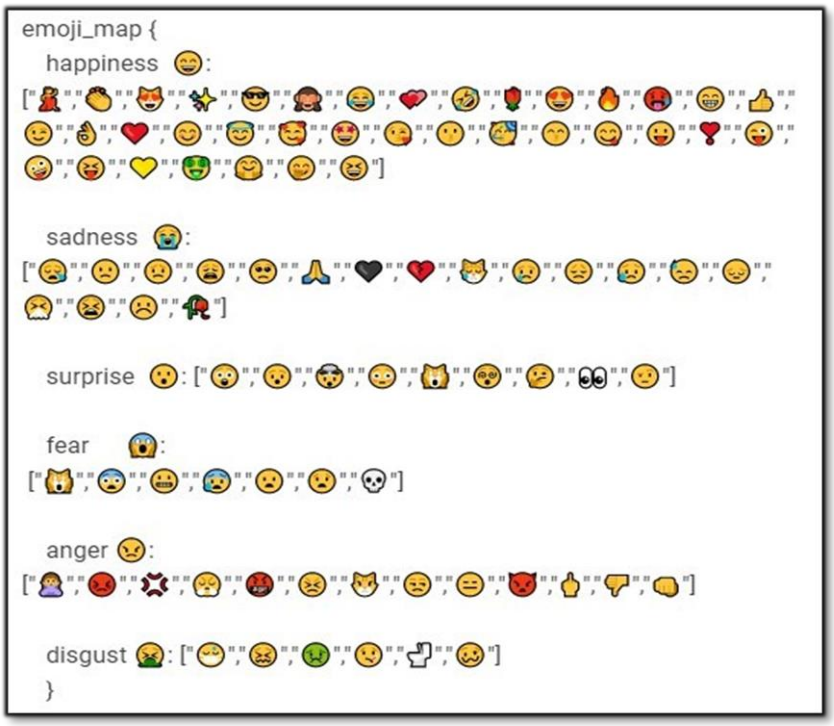



**Figure 12:** Handling emojis with emoji map.


**Figure 13:** Replace the emoji with its meaning.

**5/ Handling Repeated Characters:** Normalize repeated characters or elongated words. To reduce noise and improve text representation.

**6/ Remove stop words:** Stop words are removed from text prior to training machine learning models since they appear frequently and provide little information that may be used for clustering.

In traditional natural language processing (NLP) approaches, this step is commonly used to reduce noise and improve efficiency, for our work we used this step just to compare the performance of two techniques of text handling will be mentioned in the tokenization and embedding section.

**7/ Word stemming or Lemmatization:** Lemmatization and stemming are techniques used to reduce words to their base or root form. Lemmatization converts words to their base form (lemma), considering their part of speech, while stemming involves removing prefixes or suffixes to obtain the root form. These techniques help in reducing the dimensionality of the text data and handling variations of words.

The weakness of several stemming functions appeared special in Arabic here is some examples:

- Using ISRIStemmer function from NLTK package to stem Arabic tweet:

```
arabic tweet :  لمزيد من المعلومات مرحبا بكم تفضلوا بزيارة هذا الرابط
after removing stop words : ['لمزيد', 'المعلومات', 'مرحبا', 'تفضلوا', 'بزيارة', 'الرابط']
after using ISRIStemmer from nltk to stem the tweet: زيد علم رحب فضل زير ربط
```

**Figure 14:** ISRIStemmer Stemming.

- Using SnowballStemmer function from NLTK package:

```
arabic tweet :  لمزيد من المعلومات مرحبا بكم تفضلوا بزيارة هذا الرابط
tweet after removing arabic stop words: ['لمزيد', 'المعلومات', 'مرحبا', 'تفضلوا', 'بزيارة', 'الرابط']
After using SnowballStemmer function with parameter 'arabic': مزيد معلوم مرحب تفضل زيار رابط
```

**Figure 15:** SnowballStemmer Stemming.

- Using ArabicLightStemmer() function from tashaphyne package:

```
arabic tweet :  لمزيد من المعلومات مرحبا بكم تفضلوا بزيارة هذا الرابط
tweet after removing arabic stop words: ['لمزيد', 'المعلومات', 'مرحبا', 'تفضلوا', 'بزيارة', 'الرابط']
after using ArabicLightStemmer function  مزيد معلوم مرحب فضل زيار رابط
```

**Figure 16:** ArabicLightStemmer Stemming

### 3.5.3. Tokenization and Embedding

In this phase, we treated our data with the two following techniques in order to compare their performance with clustering, to choose the best one.

- **TFIDF (Term Frequency-Inverse Document Frequency):** Scikit-Learn provides a transformer called the TF-idf-Vectorizer for vectorizing documents with TF–IDF scores. It uses the CountVectorizer estimator to count occurrences of tokens, followed by a Tf-idf-Transformer to normalize these occurrence counts by the inverse document frequency. The input for the transformer is strings containing a collection of raw tweets.
- **BERT Model (Bidirectional Encoder Representations from Transformers):** The BERT model is a neural network architecture developed by Google for natural language processing (NLP) tasks. It is based on the transformer architecture, which is a type of deep learning model designed to handle sequential data. It is a pre-trained language model, meaning it has been trained on a large corpus of text in an unsupervised way without specific labels or annotations. By training the model on a large amount of text data, BERT learns to capture the underlying patterns and structure of language, which can then be applied to a variety of downstream NLP tasks.

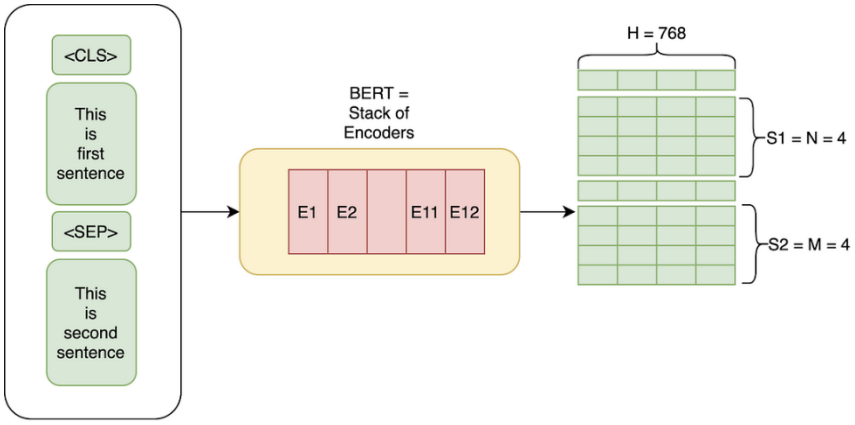



**Figure 17:** Bert Architecture.

#### 3.5.3.1. Tokenization

The tokenization is the process of splitting a text document into smaller units called tokens. In the context of the pre-trained model BERT, tokenization plays a crucial role in preparing the input text for the model. The **AutoTokenizer** class in the transformers library provides an interface to automatically select the appropriate tokenizer based on the specified pre-trained model.

The input to a BERT model would typically be the text of the tweet itself after pre-processed. Where the **encode()** method combines tokenization and conversion steps, where it takes the tweet as input and adds special tokens, such as [CLS] (classification) token at the beginning and [SEP] (separator) token at the end of the tweet (these special tokens help BERT understand the task and sentence

boundaries). It returns a list of token IDs representing the tokenized tweet, which are the basic building blocks of the BERT model's input.

```
tweet: كنت تسمع بوقتك
token_IDs: [101, 786, 31898, 766, 15450, 34783, 11693, 40218, 39053, 12497, 102]
```

**Figure 18:** Token IDs.

### 3.5.3.2. Embedding Generation

The embedding process involves transforming input text into dense vector representations called embeddings, which capture the contextual meaning of words and their relationships within a sentence.

The token IDs are fed into the BERT model, which utilizes a masked language model and a next-sentence prediction task during pre-training to learn contextual representations. The main output of the BERT model is the contextualized embeddings of the input tokens, which contain valuable contextual information, can be used as input to downstream classification or clustering to perform a specific NLP task, such as emotion detection or text classification.

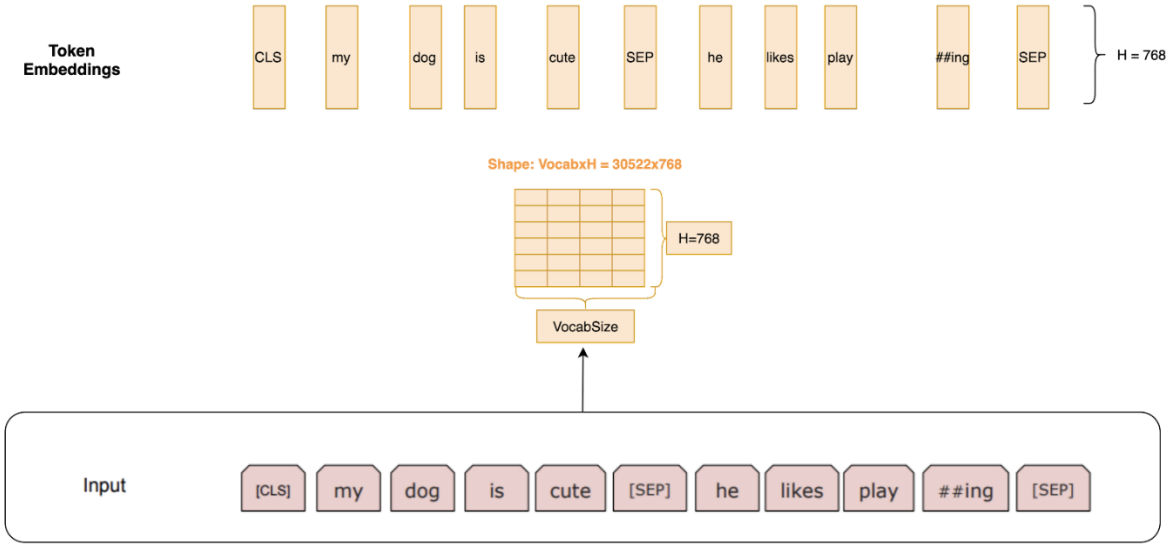



**Figure 19:** Token Embedding.

When working with models like BERT it is generally not necessary or recommended to perform extensive pre-processing steps like stop word removal or lemmatization because the pre-trained model Bert is designed to capture the contextual information of words and subwords within a sentence. It utilizes a token-level modelling approach, where each subword is assigned its own embedding. This allows BERT to capture fine-grained relationships and dependencies between

subwords, which helps it understand the context and meaning of the text. Therefore, in this situation, we did not remove stop words or stem words in data pre-processing step.

### 3.5.4. Dimensionality Reduction

Datasets often contain a large number of features, which can lead to challenges such as increased computational complexity, overfitting, and difficulty in visualization. Principal Component Analysis (PCA) is one of the commonly used techniques for dimensionality reduction, which aim to address these challenges by identifies the directions, known as principal components, along which the data varies the most, and projecting the data onto a subset of these principal components.

### 3.5.5. Clustering

The use many different clustering techniques allow us to study their grouping strategy and their effectiveness. We created a labeled Algerian dataset based on the clustering technique, where we described every used technique in this section.

#### 3.5.5.1. Standard Algorithms

To group similar data points together based on their inherent patterns or similarities, we used the following standard algorithms for select the suitable for our data, especially because of the tweets with informal language on our dialectical Algerian data.

**1/ K-means Algorithm:** K-means is a computationally efficient and easy-to-implement algorithm. K-means algorithm aims to partition a dataset into K distinct clusters. This technique showed great results in text clustering. The defining of K-parameter, help us specify the number of clusters which represents the six emotions in Ekman emotional models (happy, sad, fear, disgust, surprise, anger) selected in the task of emotions detection.

**3/Agglomerative Clustering:** This technique is based on the differences between the objects to be grouped together. A particular type of dissimilarity may be appropriate for the subject being examined and the nature of the data, but the time and space complexity of agglomerative clustering is more than K-means clustering.

**2/Gaussian Mixture Algorithm:** Gaussian mixture models classify data using probability distribution. Mixture models have the advantage of not requiring which subgroup a data point belongs to. It enables the model to automatically learn the subpopulations.

The studies and the previous related works mentioned indicated that the ensemble clustering technique give a great result, but before adopting this technique, we must improve that it give a better result than

a standard algorithm. where we used the ensemble clustering which combines two main algorithms K-means and Gaussian Mixture with IRIS data set [59], clearly showed the improvement of the performance, where this result will be mention in the results section.

#### 3.5.5.2. Ensemble Clustering

There are different methods to create an ensemble clustering. One common approach is to use consensus functions. We provided an ensemble clustering by combined the two algorithms K-means and Gaussian Mixture after that we aggregate the results by majority voting technique. The algorithms selection was based on many reasons that we can illustrate:

- ✓ Even with the good clustering of K-means algorithm's the performance can be sensitive to the initial choice of cluster centers, and it may converge to local optimal. To minimize the consequences of this problem, we combine it with the Gaussian Mixture algorithm.
- ✓ In additional, the previous standard algorithms trying step showed the performance of the algorithms showing in the result section with our data, make us confirm that the clustering of those techniques give a great result.
- ✓ Finally, we can say that the ensemble clustering leverages the diversity and complementary strengths of different clustering approaches to overcome the limitations of individual algorithms.

**1/ Combines Multiple Clustering Algorithms**

The idea is to leverage the strengths of the k-means and Gaussian Mixture algorithms to improve the overall clustering performance. Multiple clustering algorithms combination steps:

- ➢ **Algorithm Selection**: Ensemble clustering begins by selecting the mentioned clustering algorithms. These algorithms can differ in their underlying principles, assumptions, or optimization criteria.
- ➢ **Individual Clustering**: Each base clustering algorithm generates its own clustering results. These individual clustering capture different aspects of the data based on the algorithm's unique characteristics.
- ➢ **Consensus Building**: The cluster assignments from the individual clustering are combined to create a consensus clustering. Various methods, such as majority voting, employed to determine the final cluster assignments.

**2/ Multiple Runs of K-means Algorithm**

Multiple runs of the K-means algorithm refer to the process of running the K-means clustering algorithm multiple times with randomly centroid initialization. Multiple runs of K-means algorithm steps:

- **Iteration**: The K-means algorithm is run K times, with each iteration using the data, with the random initialization of the cluster centers.
- **Assignment and Update Steps**: The K-means algorithm proceeds with the assignment step, where each data point is assigned to the nearest cluster center, and the update step.
- **Convergence**: The assignment and update steps are repeated until convergence, where the cluster assignments no longer change significantly or a maximum number of iterations is reached.
- **Selection of Final Clustering**: The final clustering solution is typically selected based on aggregate the results from multiple runs, and use majority voting to choose the results more repeated in those runs results.

### 3.5.6. Evaluation Metrics

Clustering is an unsupervised learning approach that is used to discover similarities between data items that do not have associated class labels. however, give a fair method for determining the reliability of models. Several ways have been devised to solve this, including: internal evaluation, external evaluation (based on data that is not clustered, such as external benchmarks). In addition, manual evaluation (a human specialist does manual evaluation).

**1/ Internal Evaluation:** based on clustered data and includes determining inter- and intra-cluster distances. A model is given the highest score if there is a high similarity between inter-cluster points and a low similarity between intra-cluster points. In our case, we have chosen the internal evaluation metrics follows:

- **Silhouette:** is calculated for each data point using mean intra-cluster distance and mean inter-cluster distance**.**

$$\text{Silhouette Coefficient} = \frac{\mathbf{b - a}}{\mathbf{max}(\boldsymbol{a},\boldsymbol{b})}$$

a = mean distance between the current data point and all other data points in the same cluster.

b = mean distance between the current data point and all other data points in the next nearest cluster.

The silhouette coefficient varies between -1 to 1 where:

- ✓ -1 indicating that the data point isn't assigned to the right cluster.
- ✓ 0 indicating that the clusters are overlapping, and 1 indicating that the cluster is dense and well-separated (this is thus the desirable value).
- ✓ The closer the value is to 1, the better the clustering method.

- **Calinski-Harabasz (variance ratio criterion):** is the ratio between between-cluster dispersion and within-cluster dispersion for all clusters.

$$\text{CHI} = \frac{\text{trace(Bc)}}{\text{trace(Wc)}} * \frac{\text{nE}-\text{c}}{\text{c}-1}$$

Where:

c = number of cluster,

nE = size of dataset E, and

trace (Bc) = the trace of between-cluster (inter-cluster) dispersion matrix

*trace*(*Wc*) = the trace of within-cluster (intra-cluster).

The higher the CHI score, the better the clustering model.

- **Davies-Bouldin:** Davies-Bouldin index = $\frac{1}{c} + \sum_{i=1}^{c} \max(\frac{\sigma i+\sigma j}{d(ci,cj)})\ i \neq j$

Where c is the number of clusters,

ci is the centroid of cluster i,

d(ci,cj) is the distance between centroids of two clusters,

and σi is the average distance of all data points in cluster i to ci.

Models that give low intra-cluster distances and high inter-cluster distances output a low Davies-Bouldin Index. Thus, the lower the Davis-Bouldin Index, the better the model.

## 3.6. Implementation

In this section, we give a brief view of the programing environment and the programing language used in the development of our system. Moreover, we will present the implementation details of different components of our just described system.

### 3.6.1. Programing Environment

We used a number of tools to realize our system. We give a definition of materials used to provide each step of our work in the following part.

**1/ Google Colaboratory:** A cloud-based Jupyter notebook environment, also known as Google Colab [61], enables users to create, execute, and share Python code and data analysis tools. It offers free access to potent computing resources like GPU processors as well as pre-installed libraries for machine learning and data analysis like Numpy and Matplotlib. Users have the ability to make and share notebooks, which can be used for group research, instruction, or data analysis. For those who do not have access to powerful hardware or who prefer a cloud-based computing environment, Google Colab is especially helpful. Python code can be written, run, and shared using Google's cloud-based Jupyter notebook environment, also referred to as Google Colab.

**2/Python language:** Python [62] is a high-level, interpreted programming language that is widely used in various fields such as web development, scientific computing, data analysis, artificial intelligence, and more. Python has a large standard library that provides many useful modules and functions, and it also has a vast ecosystem of third-party libraries and frameworks that can be used for various tasks. The syntax of Python is relatively easy to learn and read, and its dynamic typing system allows for rapid prototyping and development. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it a flexible and versatile language. Python is one of the most popular programming languages in the world.

### 3.6.2. Data Collection

We have mentioned in the conception section the sources choosing for collect the data one of them was Twitter API. With the help of the open source Python package Tweepy, can easy access to the Twitter API and collect data from the source. Here is an overview of Tweepy package and the code used.

**Tweepy:** is an open-source Python package that gives you a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter's models and API endpoints.

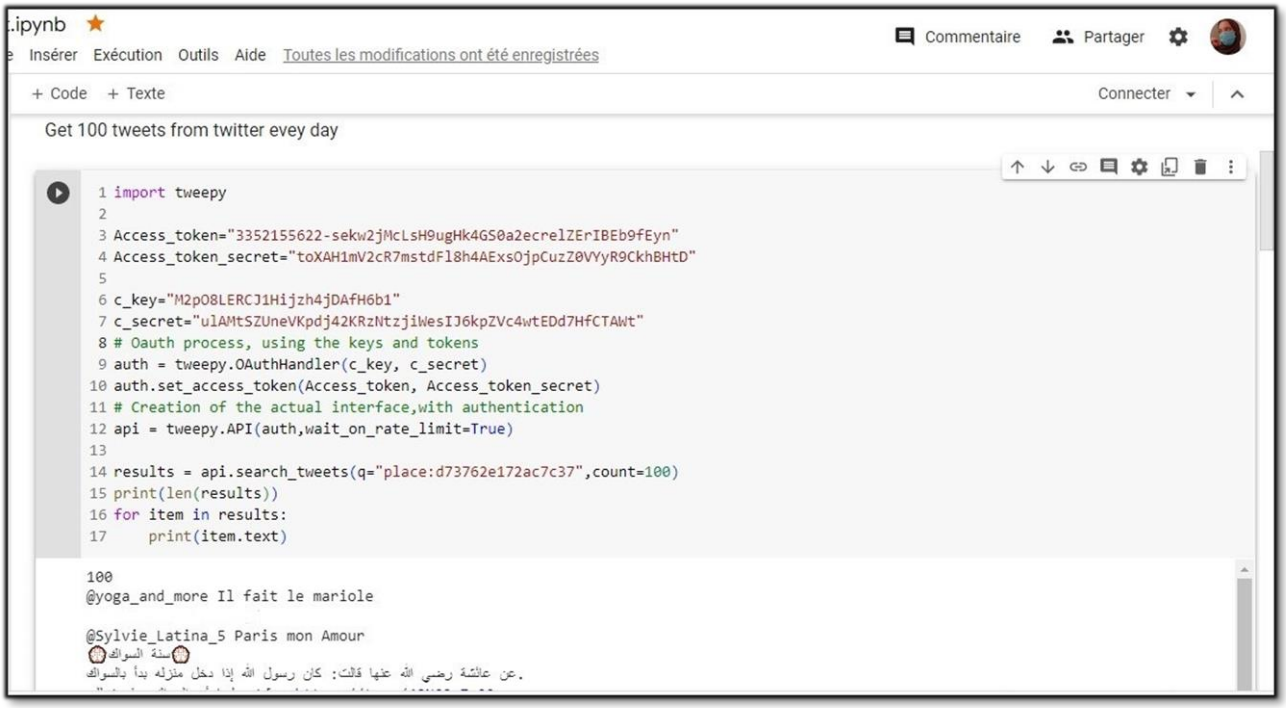



**Figure 20:** Collecting Data from Twitter API.

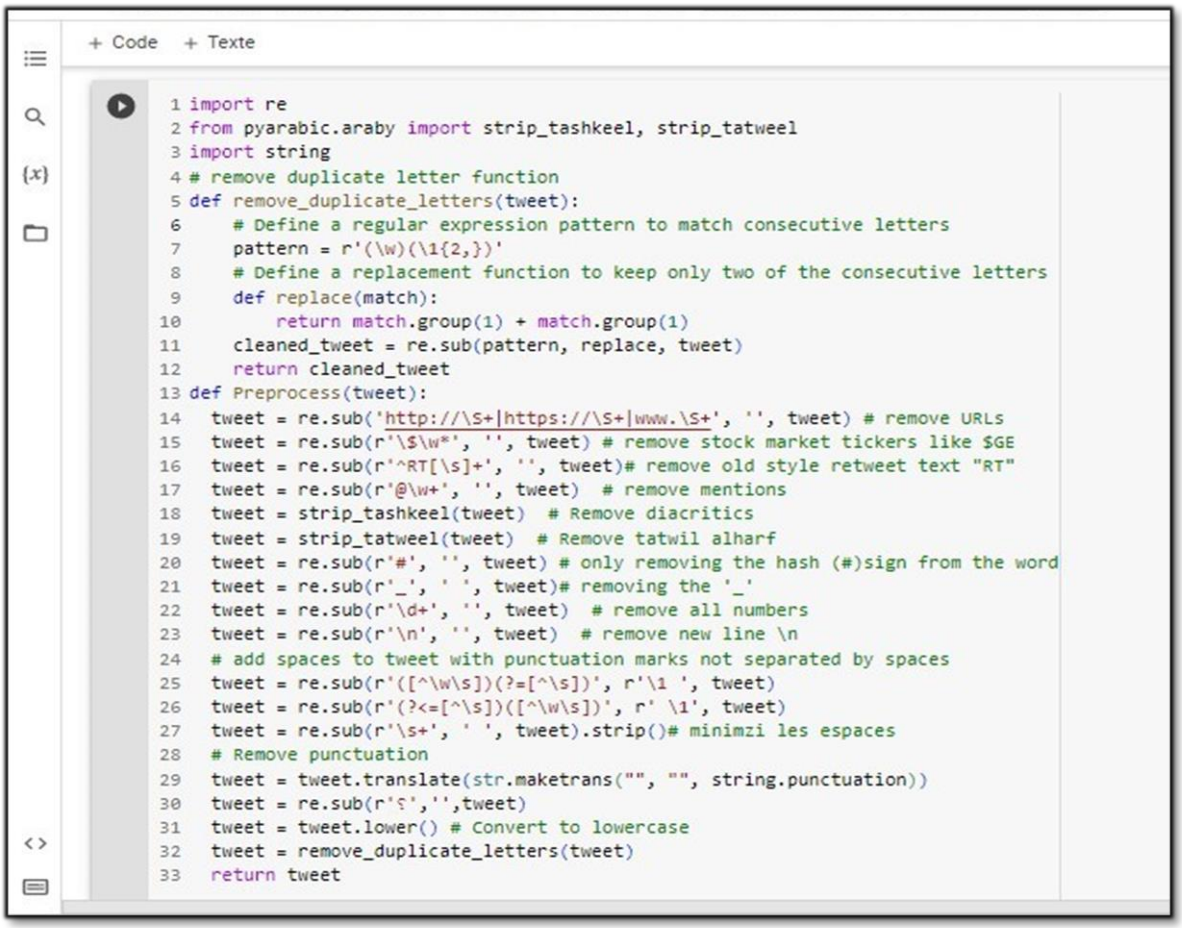
### 3.6.3. Data Cleaning

For the implementation of the pre-processed steps mentioned in our conception we used a several packages that we will described either then some of their functions that we have used.

- **PyArabic:** A specific Arabic language library for Python provides basic functions to manipulate Arabic letters and text, like detecting Arabic letters, Arabic letters groups and characteristics, remove diacritics etc. Functions were used (strip_tashkeel, strip_tatweel). [63]

- **Emoji:** The Emoji Module [64] is a Python package that enables us to use and print emojis inside of Python programs as well as in applications we are developing.

- **RegEx:** A string of characters that creates a search pattern is known as a regex, or regular expression. RegEx can be used to determine whether a string includes a given search pattern. Functions were used(re.sub, re.match).



```
+ Code + Texte

import re
from pyarabic.araby import strip_tashkeel, strip_tatweel
import string
# remove duplicate letter function
def remove_duplicate_letters(tweet):
    # Define a regular expression pattern to match consecutive letters
    pattern = r'(\w)(\1{2,})'
    # Define a replacement function to keep only two of the consecutive letters
    def replace(match):
        return match.group(1) + match.group(1)
    cleaned_tweet = re.sub(pattern, replace, tweet)
    return cleaned_tweet
def Preprocess(tweet):
  tweet = re.sub('http://\S+|https://\S+|www.\S+', '', tweet) # remove URLs
  tweet = re.sub(r'\$\w*', '', tweet) # remove stock market tickers like $GE
  tweet = re.sub(r'^RT[\s]+', '', tweet)# remove old style retweet text "RT"
  tweet = re.sub(r'@\w+', '', tweet)  # remove mentions
  tweet = strip_tashkeel(tweet)  # Remove diacritics
  tweet = strip_tatweel(tweet)  # Remove tatwil alharf
  tweet = re.sub(r'#', '', tweet) # only removing the hash (#)sign from the word
  tweet = re.sub(r'_', ' ', tweet)# removing the '_'
  tweet = re.sub(r'\d+', '', tweet)  # remove all numbers
  tweet = re.sub(r'\n', '', tweet)  # remove new line \n
  # add spaces to tweet with punctuation marks not separated by spaces
  tweet = re.sub(r'([^\w\s])(?=[^\s])', r'\1 ', tweet)
  tweet = re.sub(r'(?<=[^\s])([^\w\s])', r' \1', tweet)
  tweet = re.sub(r'\s+', ' ', tweet).strip()# minimzi les espaces
  # Remove punctuation
  tweet = tweet.translate(str.maketrans("", "", string.punctuation))
  tweet = re.sub(r'؟','',tweet)
  tweet = tweet.lower() # Convert to lowercase
  tweet = remove_duplicate_letters(tweet)
  return tweet
```

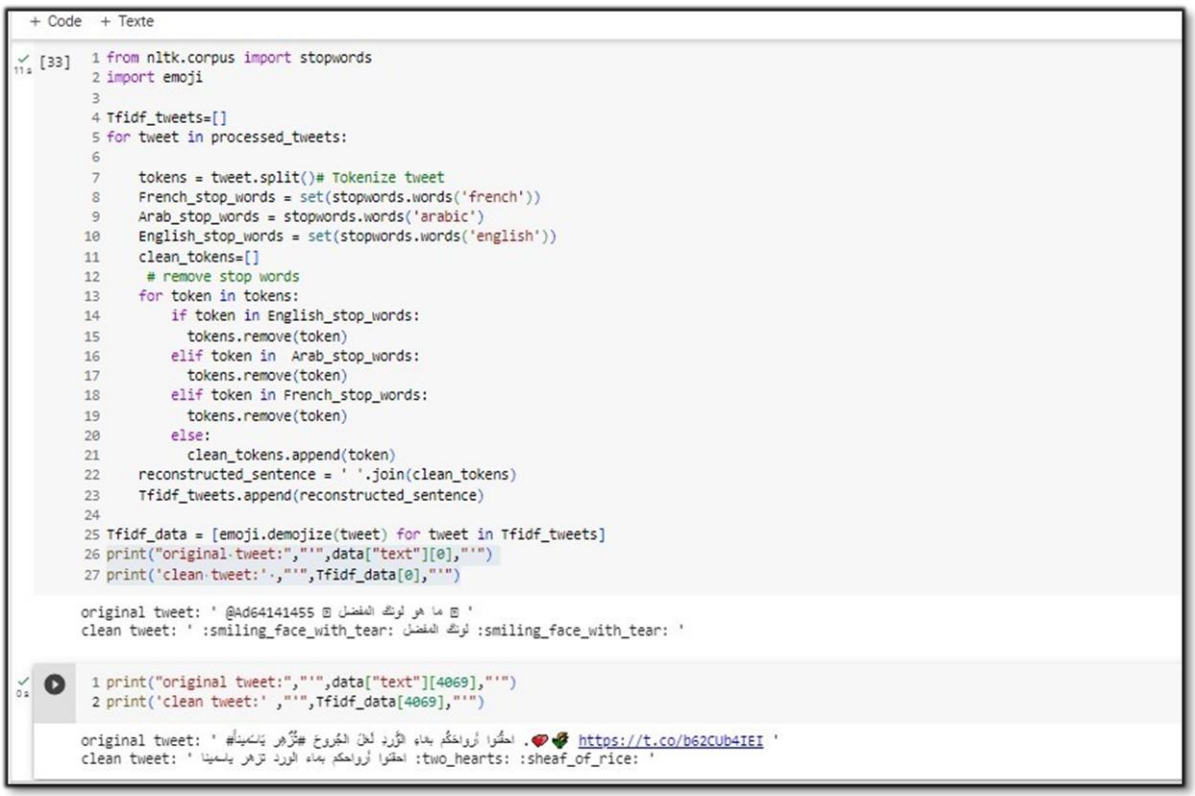
**Figure 21:** Script of Pre-processing Functions.

```
+ Code + Texte

4131                                    @qaamarihab 👍
4132          @BouthaynaBouth4 متبادلين لشخصكم الموقر
4133                                  @B123alger احسنت
Name: text, Length: 4134, dtype: object

[19]
from transformers import BertTokenizer, BertModel
import pandas as pd
import emoji
# Load tweet data and tokenize the text
data = pd.read_csv('../content/drive/MyDrive/tweets.csv')
processed_tweets = [Preprocess(tweet) for tweet in data["text"]]
cleaned_tweets = [emoji.demojize(tweet) for tweet in processed_tweets]

print(("\n").join(cleaned_tweets))

:smiling_face_with_tear: ما هو لونك المفضل :smiling_face_with_tear:
بسك ما قسنيهاش
```

**Figure 22:** Script of Read and Clean the Data.

As we mentioned before we applied both techniques TFIDF and Bert model and here are the additional preprocessed steps (removing stop words). The **Figure 23** will represent the removing of stop words from our data before the TFIDF vectorization using NLTK package.

- **Natural Language Toolkit (NLTK):** The Natural Language Toolkit (NLTK) [65] is a platform for creating Python applications for statistical natural language processing (NLP) that work with human language data. For tokenization, parsing, classification, stemming, tagging, and semantic reasoning, it includes text-processing libraries. Additionally, it comes with a cookbook and a book that explains the concepts underlying the language processing tasks that NLTK supports, as well as visual demonstrations and sample data sets. we used (stopworlds).



```
+ Code  + Texte

[33]
from nltk.corpus import stopwords
import emoji

Tfidf_tweets=[]
for tweet in processed_tweets:

    tokens = tweet.split()# Tokenize tweet
    French_stop_words = set(stopwords.words('french'))
    Arab_stop_words = stopwords.words('arabic')
    English_stop_words = set(stopwords.words('english'))
    clean_tokens=[]
     # remove stop words
    for token in tokens:
        if token in English_stop_words:
          tokens.remove(token)
        elif token in  Arab_stop_words:
          tokens.remove(token)
        elif token in French_stop_words:
          tokens.remove(token)
        else:
          clean_tokens.append(token)
    reconstructed_sentence = ' '.join(clean_tokens)
    Tfidf_tweets.append(reconstructed_sentence)

Tfidf_data = [emoji.demojize(tweet) for tweet in Tfidf_tweets]
print("original tweet:","'",data["text"][0],"'")
print('clean tweet:',"'",Tfidf_data[0],"'")

original tweet: ' @Ad64141455 ما هو لونك المفضل '
clean tweet: ' :smiling_face_with_tear: لونك المفضل :smiling_face_with_tear: '

print("original tweet:","'",data["text"][4069],"'")
print('clean tweet:' ,"'",Tfidf_data[4069],"'")

original tweet: ' احفُوا أرواحكُم بماء الوَرد لكَي تخرج #تُزهر يَاسَمينا. https://t.co/b62CUb4IEI '
clean tweet: ' احفوا أرواحكم بماء الورد تزهر ياسمينا :two_hearts: :sheaf_of_rice: '
```

**Figure 23**: Removing Stop Words before TFIDF Vectorization.

- **Scikit-Learn (Sklearn):** Sklearn (Skit-Learn) [66] is the most efficient and dependable Python machine learning library. It provides a variety of efficient tools for statistical modelling and machine learning, such as classification, regression, clustering, and dimensionality reduction, through a Python consistency interface. This library was primarily created in Python and is based on NumPy, SciPy, and Matplotlib. Functions used in our work are: TfidfVectorizer, KMeans, metrics, PCA, KFold, AgglomerativeClustering, GaussianMixture.

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Feature Extraction
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(Tfidf_data)
print(X)
```

```
(0, 4324)     0.4819463816710586
(0, 7092)     0.4819463816710586
(0, 2252)     0.7317481604953637
(1, 4779)     1.0
(2, 4103)     0.2800807655897265
(2, 6327)     0.22195801629387568
(2, 8808)     0.2800807655897265
(2, 6781)     0.2669279137524544
```

**Figure 24**: Script of TFIDF Vectorization.

### 3.6.4. Tweets Tokenization and Embedding

After using the TFIDF technique, now we will describe our choice for transform the text data to a numerical version. In the context of emotion detection, we choose from the famous platform Hagging Face the Bert Multilingual Base Model, which is one of the models in Transformer.

**Hugging Face:** is [67] a promising startup that aims to revolutionize machine learning by providing open-source natural language processing technology.

Clément Delangue and Julien Chaumond, two French engineers, launched it in 2016. Hugging Face has around 43 thousand ratings on GitHub and over 6,000 contributors. It also includes over 7,000 different models and is available in approximately 140 languages. This startup initially advertised itself as a kind of chatbot for teenagers capable of carrying on a conversation. However, its origins may be traced back to 2018, when its authors released their open-source Transformers library. This enables engineers and scientists to begin developing ML models for natural language processing.

**Sentence Transformer:** This framework [68] provides an easy method to compute dense vector representations for sentences, paragraphs, and images. The models are based on transformer networks and achieve state-of-the-art performance in various task. Text is embedding in vector space such that similar text is close and can efficiently be found using cosine similarity.

**Transformers:** Transformers [69] offers APIs that allow you to quickly download and use those pre-trained models on a given text, hone them on your own datasets, and then publish them on our model hub to share with the community.

**BERT Multilingual Base Model (cased):** [70] is a transformers model pretrained model on the top 104 languages with the largest Wikipedia of multilingual data in a self-supervised fashion. It was pretrained with two objectives: masked language modeling (MLM) and next sentence prediction

(NSP). MLM allows the model to learn a bidirectional representation of the sentence, while NSP predicts if two sentences are following each other.

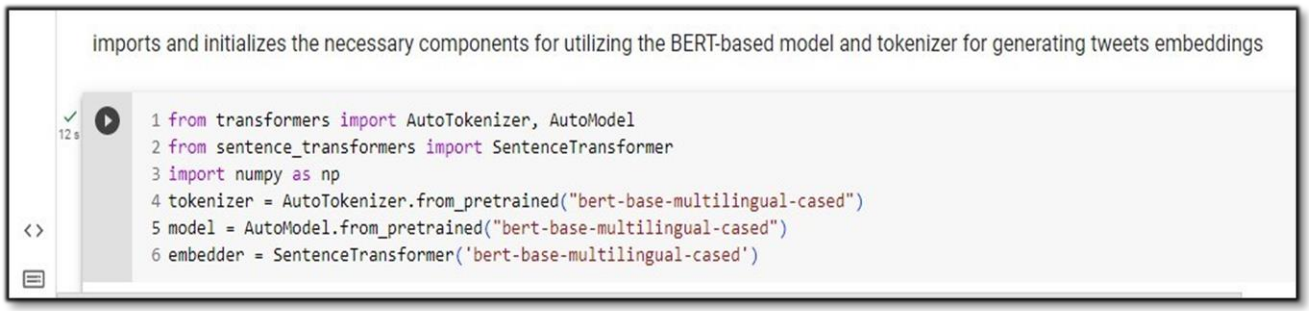



**Figure 25**: Imports and initializes BERT-based model and tokenizer

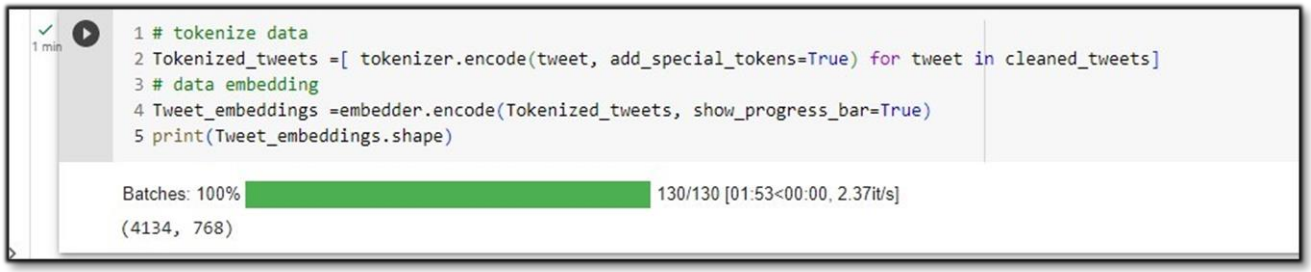



**Figure 26**: Tokens Embedding

### 3.6.5. Dimensionality Reduction

The Bert model output shape was (4134, 768), which considered a large matrix and this size make problem clustering step, for the reason that we used PCA to effectively reduce the dimensionality into (4134, 2) while retaining as much variance as possible. This allows for a compact representation of the data without losing critical patterns or structures.


**Figure 27**: PCA Tweets-Embedding Dimensionality Reduction.

### 3.6.6. Clustering

The main step in our work is clustering, where we have applied some techniques for the goal of studied their performance to can select the suitable algorithms for our data. The algorithms is mentioned in the conception step, and here we will specify the parameters used to get our result.

#### 3.6.6.1.Using Standard Algorithm

We choose (K-means algorithm, Gaussian Mixture and Agglomerative) to compare the performance of the techniques and that help selecting the better techniques in our clustering ensemble.

#### 1/ K-means Clustering

En the order to comparing the performance of the two different techniques TFIDF and BERT model, we used their outputs as a K-means algorithm input with the same following parameters:

- Number of clusters = 6
- Random state = 45
- Initialization = 'k-means++'
- Number of initialization = '7'

Where in our work the number of cluster represent the six emotion (happy, sad, anger, fear, surprise, disgust), that emotions going to be the labels predicted to each tweet in the data.

```
+ Code  + Texte

from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
# Define the label map
label_map = { 0: 'angry', 1: 'happy', 2: 'sad',3: 'fear', 4: 'surprise',5: 'disgust'}
# Initialize KMeans object
num_clusters = 6
kmeans = KMeans(n_clusters=num_clusters, random_state=45,init='k-means++',n_init=7)
kmeans.fit(reduced_features)
# Evaluate the clustering results
predicted_labels2 = kmeans.labels_
# Replace the predicted_labels with actual labels for the tweets
B_predict = [label_map [label] for label in predicted_labels2]
print(B_predict)
labels = [ 'angry', 'happy', 'sad', 'fear', 'surprise', 'disgust']
# Count the frequency of each sentiment label in the predicted_labels
unique_labels, counts = np.unique(B_predict, return_counts=True)
# Define actual_labels as the unique labels in the test set
actual_labels = np.unique(labels)
# Calculate the percentage of each sentiment label
percentages = counts / len(B_predict) * 100
# Create a bar graph to show the percentage of each sentiment label
plt.bar(unique_labels, percentages)
plt.xlabel('Emotion Label')
plt.ylabel('Percentage')
plt.xticks(unique_labels, actual_labels)
plt.show()
```

**Figure 28**: Script of K-means clustering.

### 2/ Gaussian Mixture Clustering

After the BERT model performance showing in the previous algorithm, we applied all the following clustering techniques on the BERT model output, including Gaussian Mixture with the following parameters:

n_components=6
covariance_type='tied'

```python
#gmm with descrpt + bert
from sklearn.mixture import GaussianMixture
import matplotlib.pyplot as plt
from sklearn import metrics
# Define the label map
label_map = { 0: 'angry', 1: 'happy', 2: 'sad',3: 'fear', 4: 'surprise',5: 'disgust'}
gmm=GaussianMixture(n_components=6,covariance_type='tied')
gmm.fit(reduced_features)
predicted_labels2=gmm.fit_predict(reduced_features)
# Replace the predicted_labels with actual labels for the tweets
gmm_predict = [label_map [label] for label in predicted_labels2]
print(gmm_predict)

labels = [ 'angry', 'happy', 'sad', 'fear', 'surprise', 'disgust']
# Count the frequency of each sentiment label in the predicted_labels
unique_labels, counts = np.unique(gmm_predict, return_counts=True)
# Define actual_labels as the unique labels in the test set
actual_labels = np.unique(labels)
# Calculate the percentage of each emotion label
percentages = counts / len(gmm_predict) * 100
# Create a bar graph to show the percentage of each emotion label
plt.bar(unique_labels, percentages)
plt.xlabel('Emotion Label')
plt.ylabel('Percentage')
plt.xticks(unique_labels, actual_labels)
plt.show()
```

**Figure 29:** Script of Gaussian Mixture Clustering.

### 3/ Agglomerative Clustering

The third chosen clustering algorithm is the Agglomerative clustering algorithm, it is initialized with the following parameters:

n_clusters = 6

linkage = 'single'

agglo with emoji descrpt

```python
#agglo with descrp
from sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt
emotion_mapping = {0: 'anger', 1: 'joy', 2: 'sadness', 3: 'fear', 4: 'surprise',5: 'disgust'}
# Initialize agglo object
agglomerative = AgglomerativeClustering(n_clusters=6, linkage= 'single')

# Fit the agglo model on the tweet embeddings
agglomerative.fit( reduced_features)

AG1_predict = agglomerative.labels_

plt.scatter(X[:, 0], X[:, 1], c=AG1_predict)
plt.show()
from sklearn import metrics
# Evaluate the clustering results
silhouette_score = metrics.silhouette_score( reduced_features, AG1_predict, metric='euclidean')
calinski_harabasz_score = metrics.calinski_harabasz_score( reduced_features, AG1_predict)
davies_bouldin_score = metrics.davies_bouldin_score( reduced_features, AG1_predict)

# Print internal evaluation metrics
print("agglomerative Internal Evaluation Metrics:")
print("Silhouette Score:", silhouette_score)
print("Calinski-Harabasz Score:", calinski_harabasz_score)
print("Davies-Bouldin Score:", davies_bouldin_score)
```

**Figure 30**: Agglomerative Clustering code.

### 3.6.6.2. The Ensemble Clustering Method

There are several ways to apply ensemble clustering; therefore, we have applied to forms of our data with it to see which one gives a better result.

- **The subset form:** to split our data into subset we used KFold function from NLTK package, where it divides the data into train indexes and test indexes, then we used the train indexes as a subset to fit it in our ensemble clustering.
- **Original data form:** we used our data after reducing their dimensions in the previous step.

The figure represents our ensemble clustering class where we defined a three main functions (init, fit and predict). **Init():** to put the number of data splits, the cluster number and the estimators that will contain the models fit. **Fit():** the function has the clustering algorithm models. **Predict():** has the majority voting technique to choose final prediction.

```python
from sklearn.cluster import KMeans
from sklearn.model_selection import KFold
import numpy as np
from sklearn.mixture import GaussianMixture

class EnsembleClustering:
    def __init__(self, n_clusters=6, n_splits=22):
        self.n_clusters = n_clusters
        self.n_splits = n_splits
        self.estimators = []

    def fit(self, embeddings, y=None):
        kf = KFold(n_splits=self.n_splits)
        embeddings_splits = []
        for train_index, test_index in kf.split(embeddings):
            embeddings_splits.append(embeddings[train_index])
        for i in range(self.n_splits):

          if (i%3==0):
              kmeans=GaussianMixture(n_components= self.n_clusters,covariance_type='tied')
          elif(i%3==1):
              kmeans = KMeans(n_clusters=self.n_clusters,n_init=3, init='k-means++',random_state=45)
          else:
              kmeans=GaussianMixture(n_components= self.n_clusters,covariance_type='diag')
          #X_subset = embeddings_splits[i]
          kmeans.fit(embeddings)
          self.estimators.append(kmeans)

    def predict(self, X):
        cluster_assignments = np.zeros((X.shape[0], self.n_splits))
        for i, estimator in enumerate(self.estimators):
            cluster_assignments[:, i] = estimator.predict(X)
        return np.apply_along_axis(lambda x: np.bincount(x.astype(int)).argmax(), axis=1, arr=cluster_assignments)
```

**Figure 31:** Ensemble clustering class.

We create an object from the ensemble clustering class and use the following parameters

Number cluster = 6.

Number splits = 22.

Where the number of splits, used also as a parameter in the loops '**For**' to indicate the times of the algorithms running. The **figure 32** will shows the using of our ensemble clustering with the data and the prediction of labels, finally providing the result showed in results part.

```
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt
X, y = make_blobs(n_samples=4134, centers=6, random_state=45)
embeddings = features2 # our data
# create an object from ensemble clustering class
ensemble_clustering = EnsembleClustering(n_clusters=6, n_splits=22)
# fit the ensemble clustering
ensemble_clustering.fit(embeddings)
# predict the labels
pred = ensemble_clustering.predict(embeddings)
print(pred)
# Mapping cluster labels to emotions
emotion_mapping = {0: 'anger', 1: 'joy', 2: 'sadness', 3: 'fear', 4: 'surprise',5: 'disgust'}
# Print predicted labels for each tweet
for i, label in enumerate(pred):
    emotion_label = emotion_mapping[label]
    #print(f"Tweet {i + 1}: Predicted emotion - {emotion_label}")
plt.scatter(X[:, 0], X[:, 1], c=pred)
plt.show()
labels = [ 'angry', 'happy', 'sad', 'fear', 'surprise', 'disgust']
# Count the frequency of each sentiment label in the predicted_labels
unique_labels, counts = np.unique(pred, return_counts=True)
# Define actual_labels as the unique labels in the test set
actual_labels = np.unique(labels)
# Calculate the percentage of each sentiment label
percentages = counts / len(pred) * 100
unique_labels, counts = np.unique(pred, return_counts=True)
print("Unique Cluster Labels:", unique_labels)
print("Cluster Label Counts:", counts)
# Create a bar graph to show the percentage of each sentiment label
plt.bar(unique_labels, percentages)
plt.xlabel('Emotion Label')
plt.ylabel('Percentage')
plt.xticks(unique_labels, actual_labels)
plt.show()
```

**Figure 32:** Create an instance of ensemble class and fit the data.

### 3.6.7. Results and Discussion

This section shows all clustering results obtained after using each of clustering techniques with handling text techniques and two data used ISRI and the Algerian dialect data with the two forms of it.

#### 3.6.7.1. Ensemble Clustering with IRIS Dataset Result

The result of using clustering ensemble with the following parameters to cluster IRIS dataset elements:

**Gaussian-Mixture-1** (n_components=3, covariance_type='tied')

**k-means** (n_clusters=3,n_init=7, init='k-means++', max_iter=400, random_state=42)

**Gaussian-Mixture-2** (n_components=3, covariance_type='diag'))

After comparing the result with standard Gaussian Mixture, the result was as follow:

| **Clustering Techniques:** | Internal Clustering Evolution | | |
|---|---|---|---|
| | **Silhouette Score** | **Calinski-Harabasz Score** | **Davies-Bouldin Score** |
| **GM1** | 0.49044386852357696 | 461.4088228376883 | 0.7896209904546088 |
| **Ensemble Clustering (GM1,k-means, GM2)** | 0.551191604619592 | 561.5937320156642 | 0.6660385791628493 |

**Table 4**: Results of clustring IRIS dataset.

### 3.6.7.2. Ensemble Clustering on Algerian data results

In this section we showed our clustering ensemble result with the Algerian dataset where we created automatically the labeling for the data tweets.

We provided a comparative table showing the internal results of each used clustering technique and their parameters:

| ***The studied cases:*** | ***Accuracy with emojis description*** | | | ***Accuracy with emojis map & meaning*** | | |
|---|---|---|---|---|---|---|
| | **M1** | **M2** | **M3** | **M1** | **M2** | **M3** |
| **Tfidf + KMeans**(n_clusters=6,random_state=45,n_init=7,init='k-means++') | **0.07067392360299216** | **33.45492080221989** | **3.333500937935875** | **0.087919856531304 38** | **51.29229431687469 5** | **1.60859125077198** |
| **Bert multilingual + Kmeans**(num_clusters = 6 , random_state=45,init='k-means++',n_init=7) | **0.64028543** | **19071.79067900869** | **0.5187492592930972** | **0.71256757** | **20697.50953327363 5** | **0.45269602533241843** |
| **Kmeans clustering ensemble** (n_clusters=6, n_splits=22) **[2*KMeans**(n_clusters=6,int='k-means++', n_init=7, random_state=45)**]** **KMeans**(n_clusters=6,int=center, n_init=7, random_state=45)**]** | **0.64028543** | **19071.79067900869** | **0.5187492592930972** | **0.7125677** | **20697.51820885732 4** | **0.45269591604225073** |
| **GaussianMixture**(n_components=6 covariance_type='tied') | **0.64675444** | **18699.843840874786** | **0.4688000714948668** | **0.6113373** | **19991.50577307981 4** | **0.68829066172393 06** |
| **Ensemble k-means & mixture**(n_clusters=6, n_splits=22)[**GaussianMixture**(n_components=6,covariance_type='diag') **KMeans**(n_clusters=6, init='k-means++', random_state=45)**Gaussian Mixture**(n_components=6,covariance_type='tied')**]** | **0.6403089** | **18639.61556584963** | **0.5283201620426722** | **0.7125677** | **20697.51820885732 4** | **0.45269591604225073** |
| **Agglomerative**(n_clusters=6, linkage='ward') | **0.62610304** | **6401.815146773558** | **0.27948264369666925** | **0.63872033** | **6916.871101134495** | **0.27522363332070704** |
| **Using ensemble clustering with subset of data** | **0.45608655** | **7608.8210511151965** | **0.9123559955574664** | **0.46812522** | **4977.340686087682** | **0.7714450989064536** |

**Table 5**: Different clustering techniques results.

**M1:** ***Silhouette Score***

**M2:** ***Calinski-Harabasz Score***

**M3:** ***Davies-Bouldin Score***

In the two figures bellow show the graphical representation of our Algerian clustering result with ensemble clustering.

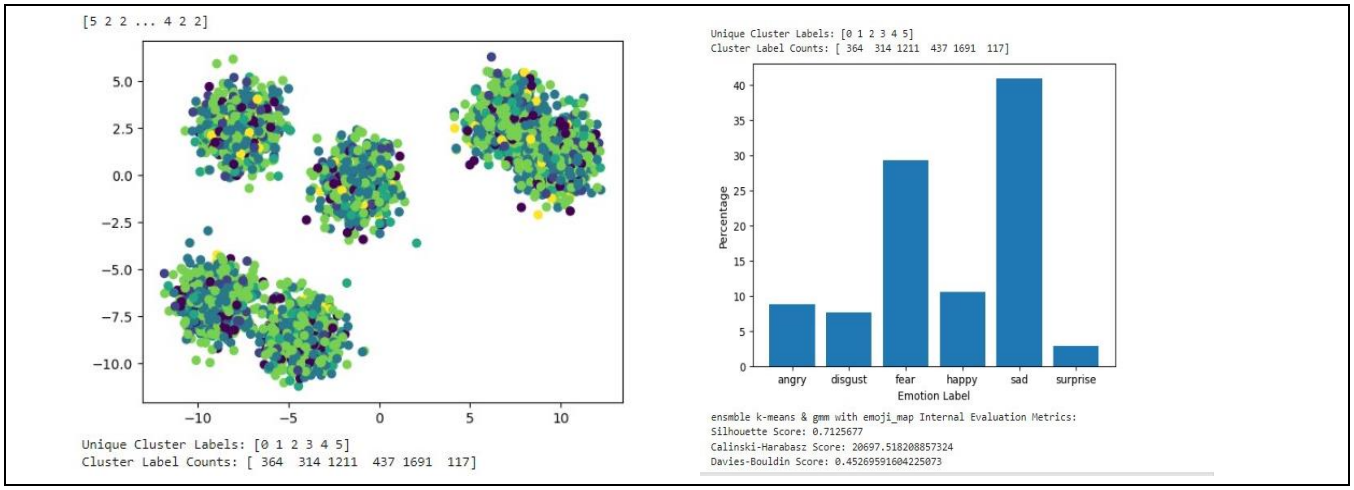



**Figure 13:** Ensemble clustering result.

## 3.7. Conclusion

Using Ensemble Clustering technique represent a number of challenges, the selection of base clustering algorithms is a key factor affecting clustering results, in our case are k-means with Gaussian Mixture. In additions, the determination of the number of clusters and the combination of individual clustering which require careful consideration. Moreover, ensemble clustering can be computationally demanding due to the need for multiple clustering runs and the integration of results. The use of Bert multilingual model with dialectical language give us a great improvement in clustering result, also handling emojis with an emoji-map and their meaning is better than other techniques especially for this type of tasks. In this section, we tried to explain different details related to our utilization of the clustering ensemble technique in the emotion detection field.

In the next section, we will present our related conclusions and our perspective in the context.

# General Conclusion

The use of social media platforms yields to the creation of a huge amongst of data which serves researchers us a source of information's to build intelligent systems able to analyze different parts of the humans lives like the behavior and emotional situations.

Many analytics tasks need a labeled data, which is rare in the Arabic language and especially in the dialectical ones. This leads us to the development of an ensemble clustering approach for automatic labeling of text data that can be helpful in several areas where this approach was able to appear its strength in many studies.

The choice of clustering technique is a key factor affecting clustering results. In our work, we present an ensemble clustering that combined multiple clustering algorithms to produce a single prediction, using a pre-trained BERT model to capturing the nuanced emotions expressed in Algerian tweets, in the context of creating an automatic labeling data.

The results derived from this work is that there is a big need to the implications for many techniques in order to well extract and understand public emotions.

For future researches, we see that some of Arabic stemming packages have to be developed to help handling Arabic text, also we hope that the dialectical language text will be highlighted and create a number of works on it to can use the results in helpful tasks. We hope also that mare ensemble clustering techniques will be used in the future to improve the process of automatic labeling methods.

# Bibliographie


[1] Blackwell, «Healthy and Unhealthy Emotion Regulation:Personality Processes ,Individual Deffernces,and life Span devlopment» *Journal of Personality*, December 2004.

[2] Damasio, Antonio .R, «Emotion in the perspective of an integrated nervous system»Department of Neurology, UniÕersity of Iowa College of Medicine, 200 Hawkins DriÕe, 1998, pp. 83–86.

[3] Armin Seyeditabari, Nagres Tabari, Wlodek Zadrozny, «Emotion detection in text» ARXIV:1806.00674v1, June 2018.

[4] P, Ekman, «Basic emotions. In Handbook Cognition and Emotion», 1999.

[5] Plutchik, Robert, «A general psychoevolutionary theory of emotion. In Theories of emotion», Elsevier, 1980, pages 3–33.

[6] Andrew Ortony, Gerald L Clore, and Allan Collins, «The cognitive structure of emotions» *Cambridge*, 1999.

[7] Russell, James A, « A circumplex model of affect» *Journal of personality and social psycholog*, p. 39, 1980.

[8] Mehrabian, James A Russell and Albert, «Evidence for a three-factor theory of emotions» *Journal of research in Personality*, pp. 11(3):273–294, 1977.

[9] Leila Kerkeni, Youssef Serrestou, Mohamed Mbarki,Kosai Raoof,Mohamed Ali Mahjoub, Catherine Cleder, «Automatic Speech Emotion Recognition Using Machine Learning» Social Media and Machine Learning, IntechOpen, 2022.

[10] Schirmer .A, Adolphs .R «Emotion detection perception from face ,voice ,and touch: comparison and convergence» Trends in cognitive sciences, pp. 216-228, 2017.

[11] Edward Cho -Chun Kao Chun-chieh Lieu, Tingo-Hao Yang, Chang -Tqi Hsieh,Von-wun Soo ,«Towards Text Based emotion Detection a survey and possible improvements» 21 may 2009.

[12] C. Cherry, S. M. Mohammad, and B. De Bruijn, «Binary classi-fiers and latent sequence models for emotion detection in suicide notes» *Biomedical informatics insights*, vol. 5, pp. 147–154, 2012.

[13] «SWISS CENTER FOR AFFECTIVE SCIENCES» [En ligne]. Available: https://www.unige.ch/cisa/.[Accès le 20 03 2023].

[14] «paperswithcode» [En ligne]. Available: https://paperswithcode.com/dataset/isear. [Accès le 10 04 2023].

[15] Alexandra Balahur, Jesu´s M. Hermida, and Andre´ s Montoyo, «Bulding and Exploting EmotiNET,a Knolwledge Base for emotion Detection Based on the apprasial theory Model» *IEEE TRANSACTION ON AFFECTIVE COMPUTING*, vol. 3, n°1, 2012.

[16] Carol Strepparava, Rada mihalaca, «SemEval-2007 Task 14: Affective text», pp. 70–74, Prague, 2007.

[17] Dan Ovedotter Alm, Dan Roth , Richard Sport, «Emotion from text: machine learning for text - based emotion detection», pp. 579–586, Vancouver, 2005.

[18] D.Turney, Saif M.Mohammad and Peter «Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon», pp. 26–34, California, 2010.

[19] Carlo Strapparava, Alessandro Valitutti, «WordNet-Affect: an Affective Extension of WordNet» Italy, 2004.

[20] Haji Binali, Chen Wu,Vidyasagar Potdar «Computational Approaches for Emotion Detection in Text» *IEEE International Conference on Digital Ecosystems and Technologies*, vol. 4, pp. 172–177, 2010.

[21] Kirk Roberts, Michael A. Roach, Joseph Johnson, Josh Guthrie, Sanda M. Harabagiu, «EmpaTweet: Annotating and Detecting Emotions on Twitter», 2012.

[22] Shuai Yuan, Huan Huang and Linjing Wu, «Use of Word Clustering to Improve Emotion Recognition from Short Text» *Journal of Computing Science and Engineering*, vol. 10, n°4, pp. 103–110, December 2016.

[23] Larissa Hjorth, Sam Hinton, «Understanding social media» *SAGE Publications Ltd*, pp. 4–15, 2013.

[24] Fiona Maclean, Derek Jones, Gail Carin-Levy, Heather Hunter, «Understanding Twitter» *British Journal of Occupational Therapy*, pp. 295–298, 2013.

[25] O'Connor Joseph, «NLP WORKBOOK», LONDON: Thorsons editon, 2001.

[26] «Alexe» [En ligne]. Available: https://www.alexe.com.

[27] «Google Home» [En ligne]. Available: https://home.google.com.

[28] «Siri» [En ligne]. Available: https://www.apple.com..

[29] Samuel Arthur L ,«Some Studies in Machine Learning Using the Game of Checkers» Originally published in IBM Journal, vol. 3, 3 July 1959.

[30] Tom M. Mitchell, «Machine Learning» McGraw-Hill Science/Engineering/Math, 1997.

[31] Li Kanghua, Jiang Shan, «Machine Learning and Cultural Production Reform——Based on the Perspective of the Development of AI Technology» *Journal of Xiangtan University (Philosophy and Social Sciences)*, 2020.

[32] Dastan Hussen Maulud, Adnan Mohsin Abdulazeez, «A Review on Linear Regression Comprehensive in Machine» *Applied Science and Technology Trends*, vol. 01, n°04, pp. 140–147, 2020.

[33] Adnan Mohsin Abdulazeez, Maryam Ameen Sulaiman,Diyar Qader «Evaluating Data Mining Classification Methods Performance in Internet of Things Applications» *SOFT COMPUTING AND DATA MINING*, vol. 1, n° 2, pp. 11–25, 2020.

[34] Béjar Javier, «Unsupervised Machine Learning and Data Mining» Stanford,California 94305,USA, Creative Commons, 2022.

[35] Dhabaleswar K. (DK) Panda, Ammar Ahmad Awan, Hari Subramoni, «High Performance Distributed Deep Learning: A Beginner's Guide» The Ohio State University, Network Based Computing Laboratory, pp. 5–20, 2019.

[36] Leslie Pack Kaelbling, Michael L. Littman, Andrew W. Moore, «Reinforcement Learning: A Survey» *Journal of Artificial Intelligence Research 4*, pp. 237–285, 1996.

[37] Jin Wei «Research on Machine Learning and Its Algorithms and Development» *Journal of Physics: Conference Series*, 2020.

[38] J. Béjar, «Unsupervised Machine Learning» UNIVERSITAT POLITÈCNICA DE CATALUNYA, 2022.

[39] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, J&g Sander, «OPTICS: Ordering Points To Identify the Clustering Structure» Institute for Computer Science, University of Munich, Germany, 1999.

[40] K.Kameshwaran, K.Malarvizhi, «Survey on Clustering Techniques in Data Mining» *International Journal of Computer Science and Information Technologies.*, vol. 5, n°2, pp. 2272–2276, 2014.

[41] Xiaofei He, Deng Cai, Yuanlong Shao, Hujun Bao, and Jiawei Han, «Laplacian Regularized Gaussian Mixture Model» *JOURNAL OF LATEX CLASS FILES*, vol. 6, n°1, pp. 1–5, 2007.

[42] JAMES C. BEZDEK, ROBERT EHRLICH,WILLIAM FULL, «FCM: THE FUZZY c-MEANS CLUSTERING ALGORITHM» *Computers & Geosciences*, vol. 10, n°2-3, pp. 191-203, 1984.

[43] Fionn Murtagh, Pedro Contreras«Algorithms for hierarchical clustering: an overview» *John Wiley & Sons, Inc.*, vol. 00, pp. 2–6, 2011.

[44] STEFANO MONTI, PABLO TAMAYO, JILL MESIROV, TODD GOLUB, «Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data» Kluwer Academic Publishers, Manufactured in The Netherlands, USA,pp. 91–118, 2003.

[45] Tossapon Boongoen, Natthakan Iam-On, «Cluster ensembles: A survey of approaches with recent extensions and applications» *Elsevier,* pp. 1–25, 2018.

[47] Vega-Pons, Sandro «A Survey of Clustering Ensemble Algorithms» *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 25, n°3, pp. 337–372, 2011.

[48] Assia Soumeura ,Mheni Mokdadia,Ahmed Guessouma,and Amina Daoudb.«Sentiment Analysis of Users on Social Networks: Overcoming the challenge of the Loose Usages of the Algerian Dialect,» *Procedia Computer Science,* 2018.

[49] Luiz F. S. Coletta, N´adia F. F. da Silva, Eduardo R. Hruschka Estevam R. Hruschka Jrl, « Combining Classification and Clustering for Tweet Sentiment Analysis» Institute of Mathematics and Computer Science University of Sao Paulo (USP) at Sao Carlos, Brazil, 2014.

[50] Kentaro Sasaki, Tomohiro Yoshikawa, Takeshi Furuhash, «Online Topic Model for Twitter Considering Dynamics of User Interests and Topic Trends» School of Engineering Nagoya University, pp. 1977–1985, 2014.

[51] Norman Aguilar-Gallegos, Laurens Klerkx, Leticia Elizabeth Romero-García Enrique Genaro Martínez-González, Jorge Aguilar-Ávila, « Social network analysis of spreading and exchanging information on Twitter: the case of an agricultural research and education centre in Mexico», THE JOURNAL OF AGRICULTURAL EDUCATION AND EXTENSION, vol. 28, n°1, 115–136, 2022.

[52] Latonero Mark, Irina Shklovski, «Emergency Management, Twitter, and Social Media Evangelism» *International Journal of Information Systems for Crisis Response and Management,* vol. 3, n°4 , pp. 1-16, 2011.

[53] Joseph, Ferdin Joe John, «Twitter Based Outcome Predictions of 2019 Indian General Elections Using Decision Tree» 4th International Conference on Information Technology, Bangkok, THAILAND, 2019.

[54] Yalong Xie, Aiping Li , Liqun Gao , and Ziniu Liu, «A Heterogeneous Ensemble Learning Model Based on Data Distribution for Credit Card Fraud Detection» *Hindawi Wireless Communications and Mobile Computing,* vol. 2021, 2021.

[55] ONAN AYTUĞ, «Two-Stage Topic Extraction Model For Bibliometric Data Analysis Based on Word Embeddings and Clustering» *IEEE Access,* vol. 7, 2019.

[56] «Kaggel» [En ligne]. Available: https://www.kaggle.com/.

[57] [En ligne]. Available: https://www.kaggle.com/datasets/didamarouane/algerian-tweets.

[58] «Emojipedia» [En ligne]. Available: https://emojipedia.org/stats/. [Accès le 10 03 2023].

[59] «sckit learn» [En ligne]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html.

[61] «colab» [En ligne]. Available: https://colab.research.google.com/notebooks/intro.ipynb. [Accès le 2023 03 12].

[62] [En ligne]. Available: https://www.python.org.

[63] [En ligne]. Available: https://pypi.org/project/PyArabic/. [Accès le 12 02 2023].

[64] [En ligne]. Available: https://pypi.org/project/emoji/. [Accès le 01 02 2023].

[65] [En ligne]. Available: https://www.nltk.org.

[66] [En ligne]. Available: https://scikit-learn.org.

[67] «Hagging Face» [En ligne]. Available: https://huggingface.co/. [Accès le 23 02 2023].

[68] [En ligne]. Available: https://pypi.org/project/sentence-transformers/. [Accès le 10 03 2023].

[69] [En ligne]. Available: https://pypi.org/project/transformers/. [Accès le 02 2023].

[70] [En ligne]. Available: https://huggingface.co/bert-base-multilingual-cased. [Accès le 03 2023].