

ALGERIAN DEMOCRATIC AND POPULAR REPUBLIC
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

KASDI MERBAH UNIVERSITY OUARGLA
FACULTY OF NEW INFORMATION AND COMMUNICATION TECHNOLOGIES
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY



THESIS SUBMITTED IN CANDIDACY FOR A MASTER DEGREE IN COMPUTER SCIENCE, OPTION
FUNDAMENTAL COMPUTING

BY MESSAOUDI NOUR ELHOUDA & YOUMBAI DIKRA LOUIZA

THEME

A CLUSTERING TECHNIQUE FOR EMOTION DETECTION FROM TEXT

EVALUATION DATE: 24/06/2024

JURY MEMBERS:

DR.	MAHDJOUR BACHIR	JURY CHAIR
DR.	MEZATI MESSAOUD	SUPERVISOR
MRS.	SAADI Wafa	CO-SUPERVISOR
DR.	ZERDOUMI OUSSAMA	EXAMINER

ACADEMIC YEAR: 2023/2024

Acknowledgments

First and foremost, all praise and thanks to Allah Almighty. We are immensely grateful to **Dr.Mezati Messaoud**, whose guidance, advices, and support were instrumental in helping us achieve our goals. His constant encouragement carried us through every stage of this work. We would also like to express our heartfelt gratitude to **Mrs.Saadi Wafa** for her advices and support throughout this work. Her contributions were significant and helpful. Finally, we extend our deepest appreciation to our parents, siblings, and the families of **Messaoudi** and **Youmbai** for their unwavering support during this academic journey and throughout our entire lives. Their encouragement and belief in us made this achievement possible.

Dedication

قال رسول الله صلى الله عليه وسلم: لا يَشْكُرُ الله مَنْ لا يَشْكُرُ النَّاسَ
صدق رسول الله صلى الله عليه وسلم

الحمد لله على إحسانه و الشكر له على توفيقه وإمّنتانه و نشهد أن لا إله إلا الله وحده لا شريك
له تعظيماً لشأنه و نشهد أن سيدنا و نبينا محمد عبده و رسوله الداعي إلى رضوانه صلى الله عليه و
على آله و أصحابه و أتباعه و سلم

بعد شكر الله سبحانه و تعالى على توفيقه لنا لإتمام هذا البحث المتواضع أتقدم بمجزيل الشكر إلى
الذي وهبني كل ما يملك حتى أحقق له آماله، إلى من كان يدفني قدما نحو الأمام لنيل المبتغى، إلى
مدرستي الأولى في الحياة بأبي الغالي على قلبي أطال الله في عمره؛

إلى التي وهبت فلذة كبدها كل العطاء و الحنان، إلى التي صبرت على كل شيء، التي رعتني حق
الرعاية و كانت سندي في الشدائد، و كانت دعواها لي بالتوفيق، تتبعني خطوة بخطوة في عملي، إلى
من ارتحت كلما تذكرت ابتسامتها في وجهي نبع الحنان أُمي أعز ملاك على القلب و العين جزاها
الله عني خير الجزاء في الدارين؛ إليهما أهدي هذا العمل المتواضع لكي أُدخل على قلبهما شيئاً من
السعادة

إلى إخوتي و أخواتي الذين تقاسموا معي عبء الحياة
كما أتوجه بخالص شكري و تقديري إلى صديقتي ذكرى لويزة يومبعي وإلى كل من ساعدني من
قريب أو من بعيد على إنجاز و إتمام هذا العمل.
رب أوزعني أن أشكر نعمتك التي أنعمت علي و على والدي و أن أعمل صالحاً ترضاه و أدخلني
برحمتك في عبادك الصالحين

مسعودي نور الهدى

قال الله تعالى: وَقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ

أَعْلَمَتْ أَشْرَفَ أَوْ أَجَلَ مَنْ الَّذِي ... يَبْنِي وَيُنْشِئُ أَنْفُسًا وَعُقُولًا

سُبْحَانَكَ اللَّهُمَّ خَيْرَ مُعَلِّمٍ ... عَلَّمْتَ بِالْقَلَمِ الْقُرُونَ الْأُولَى

الحمد لله رب العالمين حمدًا لشكره أداءً ولحقه قضاءً، ولحبه رجاءً ولفضله نماءً ولثوابه عطاءً.

إلى رسولنا الكريم...المصطفى الحبيب...الصادق الأمين...سيد المرسلين وخاتم النبيين.

إلى من كلله الله بالهبة والوقار، إلى من أحمل اسمه بكل افتخار، عزيزي وعزتي وعزي، بطلي الأوحده واستقامة ظهري، قدوتي الأولى ونعمتي الراحلة، رحلت وظل حبل الدعاء هو الوصل بيننا، أبي غفر الله لك، ورحمك، وأنس وحشتك، وجمعنا بك في جنته.

والدي العزيز الشهيد محمد الفاضل

إلى سندي وقرة عيني، إلى رفيقة دربي ونور عمتي، إلى حيي الأول وكلمتي الأولى، إلى من كان دعاءها سر نجاحي وحنانها بلسم جراحي زادك الله مكانة وصحة وإيماناً.

أُمِّي الْحَبِيبَةُ

إلى كتفي الثاني الذي لا يميل، عيني الأخرى التي تبكي معي، قلبي الذي يضحك لفرحي، أماني من الحيات، مأمني من الخذلان.

إِخْوَتِي وَأَخْوَاتِي هَدِيل، نورهان، بشير، صفوت حفظكم الله

إلى صديقة روحي، رفيقة طفولتي وصاحبة عمري، صديقة أباهي برفقتها السنين، تلك البعيدة عن عيني والقريبة لقلبي وأجمل صدفي، اللهم صديقتي حتى الجنة، دمت لي شيئاً جميلاً لا يتتهي.

سُكْرَتِي وَتَوَامِ رُوحِي لِيلِيَا

إلى رفيقة دراستي نور الهدى مسعودي، صديقة الصدوقة، يامن تنطبق عليها مقولة صحبة الكرام ترفع المقام، طاب العمر بك يا صديقتي وطبت لي عمرا.

إلى شخصي المفضل ع. ب، عافية قلبي وابتسامة أيامي، شكراً لأنك في حياتي، دمت لي أبداً. وفي مسك الحتام، لا يمكنني نسيان جاري، عمي وخليفة أبي، يامن جاور القلب فأحسن الحيرة، إلى من دام اقترابه بكل الخير، لطالما كنت لنا أهلاً غير الأهل، ومصدر أساسي للسند والدعم والثقة، حفظك الله وأدامك لنا ولعائلتك.

عمي حسين

يومبعي ذكرى لويضة

Abstract

Emotion detection from text is a critical area of research in natural language processing, especially with the rise of social media platforms like X (Twitter) and Facebook. These platforms generate vast amounts of short text, where users frequently express their emotions. By analyzing these brief posts, unsupervised learning techniques can identify and categorize feelings from short text. Our work delves into the significance of emojis and keywords and their influence on interpreting emotions positively on X. It considers emotion detection from text on X, focusing on both English and the Algerian dialects. By utilizing ensemble clustering methods, the research aims to automatically identify and categorize emotions according to Ekman's six basic emotions: happiness, sadness, anger, disgust, fear, and surprise. Ensemble clustering combines multiple clustering algorithms to improve the robustness and accuracy of the results, making it particularly useful for the informal and diverse nature of social media content. Our analysis shows that ensemble clustering performs better than single clustering methods. The silhouette score, a measure of clustering quality, is 0.82 for English data and 0.728 for Arabic data. Our findings suggest that ensemble clustering methods improve emotion detection in X text for both English and Algerian dialect.

Keywords : Emotion detection, Ensemble clustering, natural language processing, social media, X, emojis, keywords, Algerian dialect .

ملخص

يعد اكتشاف المشاعر من النص مجالاً بالغ الأهمية للبحث في معالجة اللغة الطبيعية، خاصة مع ظهور منصات الوسائط الاجتماعية مثل إكس (تويتر) وفيسبوك. تولد هذه المنصات كميات هائلة من النصوص القصيرة، حيث يعبر المستخدمون بشكل متكرر عن مشاعرهم. ومن خلال تحليل هذه المنشورات الموجزة، يمكن لتقنيات التعلم غير الخاضعة للرقابة تحديد المشاعر وتصنيفها من النص القصير. تتعمق هذه الدراسة في أهمية الرموز التعبيرية والكلمات الرئيسية وتأثيرها على تفسير المشاعر. وهو يبحث في اكتشاف المشاعر من النص على إكس، مع التركيز على اللهجتين الإنجليزية والجزائرية. من خلال استخدام أساليب التجميع الجماعي (Ensemble clustering)، يهدف البحث إلى تحديد العواطف وتصنيفها تلقائياً وفقاً لمشاعر إيكمان الأساسية الستة: السعادة والحزن والغضب والاشمئزاز والخوف والمفاجأة. تجمع المجموعات بين خوارزميات التجميع المتعددة لتحسين قوة ودقة النتائج، مما يجعلها مفيدة بشكل خاص للطبيعة غير الرسمية والمتنوعة لمحتوى الوسائط الاجتماعية. يوضح تحليلنا أن أداء المجموعات المجمعّة أفضل من طرق التجميع الفردية. تبلغ درجة الصورة الظلية (The silhouette score)، وهي مقياس لجودة التجميع، 0.82 للبيانات الإنجليزية و 0.72 للبيانات العربية. تشير النتائج التي توصلنا إليها إلى أن أساليب تجميع المجموعات تعمل على تحسين اكتشاف المشاعر في نص إكس لكل من اللهجة الإنجليزية والجزائرية.

الكلمات المفتاحية: معالجة اللغة الطبيعية، Ensemble clustering، وسائل التواصل الاجتماعي،

إكس، الرموز التعبيرية، اللهجة الجزائرية.

Résumé

La détection des émotions à partir de textes est un domaine de recherche crucial dans le traitement du langage naturel, en particulier avec l'essor des plateformes des médias sociaux comme X (Twitter) et Facebook. Ces plateformes génèrent de grandes quantités de textes généralement courts, où les utilisateurs expriment fréquemment leurs émotions. En analysant ces brefs messages, les techniques d'apprentissage non supervisé peuvent identifier et catégoriser les sentiments d'un texte court. Notre travail explore l'importance des emojis et des mots-clés et leur influence sur l'interprétation positive des émotions sur X. Il examine la détection des émotions à partir des textes sur X, en se concentrant à la fois sur le texte anglais et le dialecte algériens. En utilisant des méthodes de clustering par l'ensemble, la recherche vise à identifier et catégoriser automatiquement les émotions selon les six émotions de base d'Ekman : la joie, la tristesse, la colère, le dégoût, la peur et la surprise. Le clustering par l'ensemble combine plusieurs algorithmes de clustering pour améliorer la robustesse et la précision des résultats, ce qui est particulièrement utile pour la nature informelle et diversifiée du contenu des médias sociaux. Notre analyse montre que le clustering par l'ensemble est plus performant que les méthodes de clustering individuelles dans le contexte de la détection des Emotions. Le score de Silhouette, une mesure de la qualité du clustering, est de 0,82 pour les données en anglais et de 0,728 pour les données en arabe. Nos résultats suggèrent que les méthodes de clustering par l'ensemble améliorent la détection des émotions dans les textes sur X, tant pour l'anglais que pour le dialectes algériens.

Mots-clés: Détection des émotions, clustering par ensemble, traitement du langage naturel, médias sociaux, X, emojis, mots-clés, dialecte algérien.

Contents

General Introduction	1
1 Emotion Detection in social media	3
1 Introduction	3
2 Definition of Emotion	3
3 Components of Emotion	4
3.1 Physiological components	4
3.2 Cognitive components	4
3.3 Behavioural components	5
4 Types of Emotion	5
4.1 Basic Emotions	5
4.2 Complex Emotions	5
5 Emotion models	6
5.1 Categorical Emotion Models	6
5.2 Dimensional Emotion Models	7
6 Importance of Understanding Emotion	10
7 Emotion detection	10
7.1 Methods of Emotion Detection	10
7.2 Applications of Emotion Detection	11
7.3 Emotion detection vs sentiment analysis	12
7.4 Text based Emotion Detection	13
7.5 Techniques for text based Emotion Detection	13
8 Social Media and Emotion Detection	14
8.1 Applications in social media	14
8.2 X Platform	15

9	Challenges in emotion detection	15
10	Conclusion	16
2	Machine learning	17
1	Introduction	17
2	Artificial intelligence	17
3	Machine Learning	18
4	Types of Machine Learning	18
4.1	Supervised learning	18
4.2	Reinforcement Learning	20
4.3	Unsupervised Learning	20
5	Conclusion	26
3	Text Clustering	28
1	Introduction	28
2	Definition of Text Clustering	28
2.1	Long Text Clustering	29
2.2	Short Text Clustering	29
3	Applications of Short Text Clustering	29
4	Challenges in Short Texts Clustering	30
4.1	Sparse Feature Vector	30
4.2	Polysemy	30
4.3	Synonymy	31
5	Text Clustering and Emotion Detection	31
6	A Comparative Study of Text Clustering Techniques for Emotion Detection in Social Media Data	31
7	Ensemble Learning	33
8	Methods of Ensemble supervised Learning	34
9	Ensemble Clustering	35
10	Process of ensemble clustering	35
10.1	Generation Process	36
10.2	Consensus Process	38
11	Properties and Advantages of using ensemble methods over single clustering algorithms	39

12	Applications of Ensemble Clustering in Text Analysis	40
13	Challenges and Future Directions in Ensemble Clustering	40
14	Emotion Detection Related Works	41
14.1	Supervised learning techniques	41
14.2	Unsupervised learning techniques	43
15	Conclusion	45
4	Conception And implementation	46
	Conception And implementation	46
1	Introduction	46
2	Conception	47
2.1	Collect data	47
2.2	Data preprocessing	48
2.3	BERT (Bidirectional Encoder Representations from Transformers) . .	50
2.4	Dimensionality Reduction	53
2.5	Ensemble Clustering	53
2.6	Evaluation Metrics	58
3	Implementation	59
3.1	Programing Environment	59
3.2	Tweets Tokenization and Embedding	62
3.3	Dimensionality Reduction	64
3.4	Ensemble Clustering	64
3.5	Analysis of English Data	66
3.6	Analysis of Arabic Data	76
3.7	Conclusion	87
	General Conclusion	88

List of Figures

1.1	Russell’s circumplex model of effects [87]	8
1.2	Plutchik’s wheel of emotions [77]	9
2.1	Basic Architecture of Supervised Learning [27]	19
3.1	Diagram of the general process of cluster ensemble [99]	36
3.2	Diagram of the principal clustering ensemble generation mechanisms [99]	37
4.1	Basic steps of Emotion Detection in our work	47
4.2	Basic steps of data preprocessing stage	50
4.3	Bert Architecture [22]	51
4.4	Token ID	52
4.5	Embeddings [17]	52
4.6	Extract Key Words and Omit Stop Words	53
4.7	Emojis description.	54
4.8	Handling emojis with emoji map.	54
4.9	Replace the emoji with its meaning(Arabic data).	55
4.10	Replace the emoji with its meaning (English data).	55
4.11	The six basic emojis with their descriptions.	55
4.12	Sample Representative word set (English data).	56
4.13	Sample Representative word set (Arabic data).	56
4.14	Imports and initializes BERT-based model and tokenizer.	63
4.15	Tokens Embedding.	63
4.16	PCA Tweets Embedding Dimensionality Reduction.	64
4.17	Ensemble clustering class.	65
4.18	Create instance of ensemble clustering class and fit the data.	66

4.19 Exploring English Data Features: Insights into Emojis.	67
4.20 Script of Pre-processing Functions.	69
4.21 Read and Clean the Data.	69
4.22 Result of method 3.	73
4.23 Exploring Arabic Data Features: Insights into Emojis.	76
4.24 Pre-processing additional Functions.	78
4.25 Best Result of Ensemble clustering	87

List of Tables

3.1	Summary of Previous works for Emotion Recognition from Text	32
4.1	Result of KMeans clustering ensembles in English data	70
4.2	Ensemble clustering using Agglomerative and GMM in English	72
4.3	Result of Ensemble clustering in English data	74
4.4	Results of different parameter initialization in English	75
4.5	Result of KMeans clustering ensembles in Arabic data with Arabic keyword	79
4.6	Result of KMeans clustering ensembles in Arabic data with English keyword	80
4.7	Ensemble clustering using Agglomerative and GMM in Arabic data with Arabic keyword	81
4.8	Ensemble clustering using Agglomerative and GMM in Arabic data with English keyword	82
4.9	Result of Ensemble clustering in Arabic data with Arabic keyword	83
4.10	Result of Ensemble clustering in Arabic data with English keyword	84
4.11	Results of different parameter initialization in Arabic data with Arabic keyword	85
4.12	Results of different parameter initialization in Arabic data with English keyword	86

General Introduction

Social media platforms have emerged as a prominent arena where individuals express their thoughts, opinions, and emotions. These platforms generate an immense volume of textual data daily, encompassing a wide range of topics and sentiments. Leveraging this data using advanced artificial intelligence (AI) and machine learning (ML) techniques can yield valuable insights, driving advancements in various fields. One of the most significant applications of extracted social media data is in comprehending and detecting emotions, a critical aspect for understanding user emotion, behavior, and interaction patterns.

Emotion detection, also known as emotion recognition, refers to the process of identifying and categorizing emotions expressed in different kind of data including textual data. This involves determining whether text conveys specific emotions such as joy, sadness, anger, and fear. By integrating machine learning approaches with natural language processing (NLP), we can develop sophisticated models that accurately identify these emotions. NLP techniques enable the processing and analysis of large-scale textual data efficiently, facilitating numerous applications such as customer feedback analysis, mental health monitoring, and social media marketing.

To detect emotions from text, we mainly need well-trained machine learning models. Training these models necessitates labeled datasets, which are difficult to obtain and collecting them takes a lot of time and effort due to the manual and lengthy process of labeling, especially in the Algerian dialect. In this context, our primary goal is the automatic labeling of data to create datasets for emotion detection. Additionally, we address the question: "What is the importance of emojis in short text and their influence on emotion detection?" and focus on the semantic aspect, particularly keywords.

Clustering alone cannot solve these problems because it is challenging to find the best algorithm for the data, and weaknesses in individual algorithms can affect the results. Hence, relying on single algorithms like k-means or DBSCAN is insufficient. To overcome

these limitations, we employ ensemble clustering to combine the strengths of multiple clustering algorithms, ensuring better accuracy and reliability in clustering tasks.

In this work, we explore the methodologies, techniques, and challenges associated with applying machine learning techniques concentrating on text clustering for emotion detection in the context of social media mining. Text clustering is a crucial step in this process, as it groups similar pieces of text together, facilitating the identification of common themes and sentiments. We place a particular emphasis on ensemble clustering, which involves combining different object representations, algorithms, and initializations.

The resulting datasets are labeled according to the Ekman emotional model, which categorizes emotions into six basic types: happiness, sadness, fear, disgust, anger, and surprise. By employing ensemble clustering techniques, we aim to produce a labeled dataset that is well-suited for supervised learning methods. This approach is applied to both Arabic text, focusing on the Algerian dialect, and English data, ensuring a comprehensive analysis across different languages and cultural contexts.

A unique aspect of our research is the focus on handling emojis and keywords, which play a significant role in conveying emotions on social media. Emojis, in particular, are widely used to express feelings and attitudes, making their accurate interpretation essential for effective emotion detection.

The structure of this dissertation is started with Chapter one introducing various concepts related to the domain of emotion detection. Chapter two presents different aspects related to machine learning techniques. Chapter three delves into text clustering techniques, with a specific focus on ensemble clustering and related works. Finally, Chapter four describes the steps we followed to create an automatically labeled dataset using the ensemble clustering method in social media where we will discuss the tools used in this work, as well as the results and discussions stemming from our research. At the end we will finish by a general conclusion which will discuss our concluding remarks and expected perspectives.

CHAPTER 1

Emotion Detection in social media

1 Introduction

In today's digital age, social media is a prevalent platform for human interaction, offering a wealth of data that reflects different emotions. Understanding these emotions is vital for improving communication, mental health interventions and advancing technologies like artificial intelligence.

In this chapter, we will explore the context of emotion detection in social media. Starting by defining key terms and examining the various components and types of emotions. Different prominent models for classifying emotional expressions are presented. Methods and applications of emotion detection are discussed, comparing it with sentiment analysis and focusing on techniques for emotion detection from text. The specific context of emotion detection on platforms like X is also explored, highlighting the challenges faced in this field. This foundation provides a deeper understanding of how emotions are detected, interpreted, and utilized in the digital world.

2 Definition of Emotion

Several researchers in the field of emotions agree on a high-level definition of emotions that view emotions as states that reflect evaluative judgments of the environment, the self and other social agents, in light of the organism's goals and beliefs, which motivate and coordinate adaptive behaviour.

Emotion is defined by Lexical as "*A strong feeling deriving from one's circumstances, mood, or relationships with others.*" [49]

In psychology, emotion can be defined as psychological states differently connected with contemplations. It can also be defined as sentiments that result in physical changes reflect one's thoughts and conduct during that state.

Emotions are often further categorized into basic emotions and complex emotions (i.e., emotions that are hard to classify under a single term such as guilt, pride, shame, etc.).

3 Components of Emotion

Also known as elements of emotional experience, the components of emotion involve breaking down emotions into distinct, interrelated parts. These components collectively contribute to our understanding and experience of emotions. Prominent components include:

3.1 Physiological components

The physiological component [55] of emotion is bodily reactions and changes associated with different emotions, such as increased heart rate, flushed skin, tense muscles, dry mouth, etc. These reactions are governed by the endocrine, nervous, and immune systems and prepare the body for various emotional responses.

3.2 Cognitive components

Cognitive components [54] are the thoughts, mental evaluations, and appraisals that influence and shape an emotional experience. This includes assessments of situations, assignments of meaning, interpretations of events, evaluation of coping resources, etc. Cognition plays a key role in determining specific emotional responses.

3.3 Behavioural components

Behavioural components [41] are the set of outward emotional expressions and actions associated with internal feeling states. Examples include facial expressions like smiling or frowning, vocal expressions like shouting or soft speech, and body language like excited gestures or listless movements, Behaviours serve as signals to others about emotions.

Emotions are often further categorized into basic emotions and complex emotions (i.e., emotions that are hard to classify under a single term such as guilt, pride, shame, etc.).

4 Types of Emotion

The study of emotions reveals two primary categories: basic emotions and complex emotions, each with distinct characteristics and origins.

4.1 Basic Emotions

The most well-known categorization of basic emotions comes from Ekman [28] who proposed 6 primary emotions based on cross-cultural facial expressions: Anger, Disgust, Fear, Happiness, Sadness and Surprise. These emotions are considered innate, universal responses that emerge early in human development and have distinct physiological signatures and facial expressions.

4.2 Complex Emotions

Complex emotions are thought to build upon and blend with more basic emotions, adding elements of intricate cognitive appraisal [54] and evaluation that incorporate social and cultural knowledge. They are considered to be more malleable to individual differences and learning experiences such as Guilt, Shame, Embarrassment, Pride, Contempt, Awe, Amusement and Desire.

The key distinction is that basic emotions have universal signals and causes while complex emotions are more constructed through social and cultural lenses. However, there can be overlap between the two categories.

5 Emotion models

The finding that emotions are experienced and recognized by humans has influenced the way that emotions are discussed in scientific contexts. According to psychology scientists, persons have "internal mechanisms for a small set of reactions (usually happiness, anger, sadness, fear, disgust, and interest) that, once triggered, can be measured clearly and objectively." Therefore, feelings like fear, grief, and rage are viewed as entities that scientists may study. To this end, there are two unique models for signifying emotions: the categorical model and the dimensional model.

5.1 Categorical Emotion Models

Also known as discrete emotion models, the categorical model of emotions involves placing emotions into distinct classes or categories. Prominent among them include:

5.1.1 The Paul Ekman model

The Paul Ekman model [28] makes emotional distinctions using six fundamental categories. These core feelings include fear, surprise, rage, disgust, sadness, and happiness. But the combination of these feelings could also result in the production of additional complicated feelings like pride, lust, greed, remorse, humiliation, and so on.

5.1.2 The Robert Plutchik model

The Robert Plutchik model [77], like Ekman's, suggests that there are only a few basic emotions that happen in opposing pairs and combine to form complicated emotions. He listed eight of these basic emotions, which include anticipation and acceptance/trust, in addition to the six main emotions that Ekman proposed. Joy versus sadness, trust versus disgust, anger versus fear, and surprise versus anticipation are the eight emotions that are paired oppositely. Plutchik asserts that depending on how an experience interprets events, there are different levels of intensity for each emotion.

5.1.3 Orthony, Clore, and Collins (OCC) model

The Orthony, Clore, and Collins (OCC) model [71] Disagreed with the Ekman and Plutchik analogy of basic emotions. Nonetheless, they all believed that feelings varied in strength and that emotions emerged from people's perceptions of events. To cover a much wider range of emotions, they discretized emotions into 22 classes: relief, envy, reproach, self-reproach, appreciation, shame, pity, disappointment, admiration, hope, fears-confirmed, grief, gratification, gloating, like, and dislike.

5.2 Dimensional Emotion Models

A dimensional model is an additional technique for identifying emotions. It denotes effects in a dimensional form. In this approach, the different emotional states are connected by a shared set of dimensions. They can be characterized in a three-dimensional space (valence, arousal, and power) or a two-dimensional space (valence and arousal). There is a place for every feeling in this space.

5.2.1 Russell model

The Russell model [87] Offers a two-dimensional circular model known as the circumplex of effect that is well-known in the representation of dimensional emotions. The paradigm uses the Arousal and Valence domains to separate emotions into two categories: Activations and Deactivations for Arousal, and Pleasantness and Unpleasantness for Valence. Figure 1.1 illustrates how the Circumplex model of effect demonstrates that emotions are connected rather than isolated.

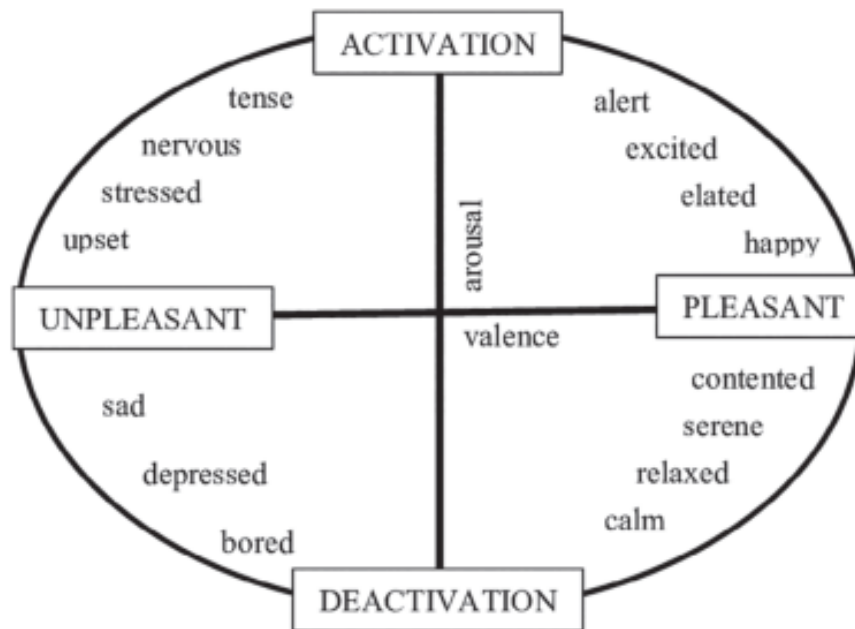


Figure 1.1: Russell's circumplex model of effects [87]

5.2.2 Plutchik model

The Plutchik model [77] displays emotions as a wheel of concentric circles, with combinations of the primary emotions on the outermost portions and the eight fundamental emotions on the innermost parts. The deepest feelings are derivatives of the eight fundamental emotions. The wheel illustrates the degree of relatedness between emotions based on where they fall on the wheel. The emotions are expressed in opposite pairs as surprise versus anticipation, joy versus sadness, anger versus fear, and trust versus disgust as illustrated in Figure 1.2.

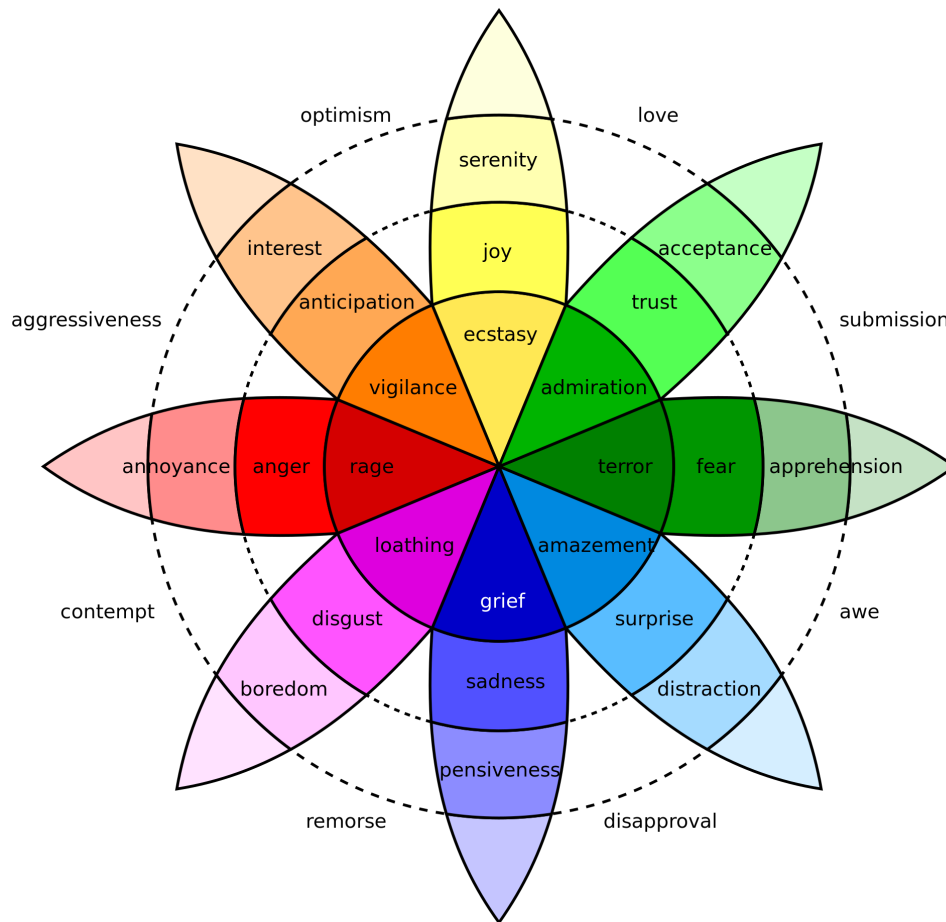


Figure 1.2: Plutchik's wheel of emotions [77]

5.2.3 Russell and Mehrabian model

The Russell and Mehrabian [88] offers a three-dimensional model of emotions as well, with the third dimension being dominance, arousal, and valence/pleasure. According to the 2-D theory, arousal and valence indicate how pleasant or unpleasant or active or passive an emotion is, respectively. The degree of emotional control that experiencers possess is characterized by the third dimension of dominance.

6 Importance of Understanding Emotion

Understanding emotion is deeply important for interpersonal communication, mental health, and artificial intelligence for the following reasons:

- In interpersonal Communication Emotion shows how people feel. Seeing each other's emotions helps people connect, respond sensitively, and bond when talking.
- Identifying emotions properly lets people look after mental health. Handling feelings well aids in coping when life is tough.
- Artificial intelligence needs to recognize human emotions, this helps AI systems act nicely, help users, and make moral choices.

In essence emotions provide key information. Understanding feelings leads to good communication, mental health help, and smart caring AI. Noticing and managing emotions thoughtfully has many benefits across those areas.

7 Emotion detection

Human been cannot exist without emotions, as emotions play a crucial role in various aspects of our lives, understanding emotions is essential because they significantly impact our mental and physical well-being, decision-making, relationships, and overall quality of life. Emotion detection (ED) or emotion recognition is a branch of sentiment analysis that deals with the extraction and analysis of emotions.[4], it refers to the use of automated methods and technologies to recognize, interpret, process, and simulate human affective states and moods [75]. Emotion detection systems aim to classify emotions such as anger, joy, sadness, fear, and disgust from multimodal data sources using machine learning techniques [28].

7.1 Methods of Emotion Detection

Emotion detection involves various methods, each leveraging different human attributes to recognize and interpret emotional states. These methods have become increasingly sophisticated with advances in technology and psychology, including:

7.1.1 Facial recognition

Facial expressions are a major way humans display emotions. Automated facial recognition [91] uses computer vision and machine learning algorithms to detect facial muscle movements, micro-expressions, etc. that signify different emotional states like happiness, sadness, anger, etc. Major techniques include facial action coding systems (FACS), neural networks, facial key point detection, etc.

7.1.2 Speech analysis

Voice tone, speech prosody, and vocal expressions also indicate emotions. This field analyses speech features [29] like pitch, intensity, and rhythm to infer affective states. Signal processing identifies acoustic patterns which are then classified using models.

7.1.3 Text analysis

Text mining tools can identify emotion from text [108] based on vocabulary, style, and context. Methods include lexicon mappings, syntactic analysis, machine learning classifiers, and hybrid approaches. Challenges involve interpreting pragmatic implications.

7.1.4 Physiological sensors

Signals [47] that detect emotional arousal via bodily signals like skin conductance, heart rate, neurological activity, etc. Multimodal frameworks combine various sensor inputs for emotion recognition. Limitations exist due to variability in responses.

7.2 Applications of Emotion Detection

Emotion detection has a wide range of applications across various fields, and its potential is constantly evolving. Here are some examples of how it's being used:

- **Customer Experience:** Analyse customer emotions, frustrations, and satisfaction from facial and vocal cues to improve product design and customer service quality.
- **Education:** Evaluate student engagement, motivation, and confusion to modify teaching methods and content delivery for better educational outcomes.

- **Automotive Safety:** Detect driver drowsiness, distraction, impairment and stress from biometrics to improve safety through alerts or autonomous aids.

7.3 Emotion detection vs sentiment analysis

Emotion detection and sentiment analysis are both essential tools in understanding human language and behavior, each serving distinct but complementary purposes in the behavioural analysis. They are related but refer to distinct techniques.

Sentiment is defined as ‘an attitude, thought, or judgement prompted by a feeling.’ Emotion, on the other hand, ‘refers to a conscious mental reaction subjectively experienced as strong feelings.’ The main difference between these two is the duration in which they are experienced [68]. That is, sentiments last for a longer period, and they are more stable than emotions [46]. Also, unlike sentiment, emotions are not necessarily targeted toward an object.

Emotion detection aims to classify specific affective states like happiness, sadness, fear, etc. conveyed in a text, audio or visual data. It relies on advanced machine learning (ML) methods to categorize precise emotion by analyzing facial expressions, speech patterns, lexical choices, etc. Emotion detection has applications in chatbots, virtual assistants, psychology, and market research. However, accurately identifying emotions is challenging given their complexity, subjectivity, and dependence on context.

In contrast, sentiment analysis focuses on determining the overall sentiment orientation as positive, negative, or neutral. It uses machine learning techniques to extract subjective information and evaluate the general attitude, appraisal, or opinion rather than pinpoint exact emotions. Common applications include social media monitoring, analyzing customer feedback, brand monitoring, and understanding public perceptions. However, sentiment analysis can miss nuances in human communication like sarcasm, and struggle to capture the intricacies of emotions.

In summary, emotion detection attempts to classify specific granular emotions while sentiment analysis categorizes the broader feeling as positive, negative or neutral. Sentiment is more enduring while emotion is intense but fleeting. Emotion can be independent but

sentiment is directed at something. Emotion analysis can be viewed as a natural evolution of sentiment analysis and its more finegrained model.

7.4 Text based Emotion Detection

Emotion detection from text, also known as text-based emotion recognition, in computational linguistics is the process of identifying discrete emotion expressed in text. [92], it is resumed in the use of natural language processing and machine learning to systematically identify, extract, and quantify affective states and subjective information from textual data it is the computational study of natural language expressed in text in order to identify its association with emotions such as anger, fear, joy, sadness, etc. [9]

Identifying a person's emotional states from a text document they have authored can be useful in a variety of contexts and domains in computational linguistics such as in e-learning environment [86] or suicide prevention [23].

7.5 Techniques for text based Emotion Detection

Emotion detection from text involves various techniques that leverage natural language processing (NLP) and machine learning algorithms to analyze textual data and infer the underlying emotions expressed within it. Some common techniques include:

7.5.1 The lexicon-based approaches

The lexicon-based [72] approaches commonly employed in emotion detection from text, making it particularly effective for identifying and categorizing emotions based on specific keywords associated with various emotional states. By utilizing established lexicons and dictionaries, this approach offers a systematic approach to mapping text to emotional categories, facilitating the analysis of emotions expressed in written content.

7.5.2 Machine Learning (ML)

Train a machine learning model on a dataset of texts labelled with emotions to build a model that can classify the emotions of new texts. We can use classical ML or deep learning

methods. Often uses linguistic features like keywords, punctuation, syntax etc in addition to word vectors. [3].

7.5.3 Deep Learning

Use deep learning neural network architectures directly on raw text or word vector inputs to classify emotion without needing to manually define features. May includes several techniques and algorithms such as CNNs, Bi-LSTMs etc.

8 Social Media and Emotion Detection

Social media [42] are online platforms and technologies that allow individuals, organizations and communities to create, share and exchange information, ideas and user-generated content. Social media platforms has become central for interpersonal communication and self-expression. Sites like Facebook [32], X (Twitter) [104], and Reddit [81] contain extensive real-time, user-generated textual data encompassing people's opinions, interests, and emotional states, this data offers a prolific resource for applying emotion detection to understand collective moods, attitudes, and emerging trends. For example, emotion detection of large-scale social data has enabled gauging public reactions to events, predicting economic indicators from collective emotions, and identifying signs of depression or self-harm risk [21]. On an individual level, users' emotional expressions on social media can be analysed with natural language processing to provide personalized recommendations, mental health interventions, or tailor-marketing. The diversity of data types - text, audio, images - also allows for detecting emotive signals from multiple modes of expression. Through automated emotion detection on social platforms, researchers gain an enriched, scalable, and frequently updated understanding of population-level behaviours and health.

8.1 Applications in social media

Emotion detection in social media has become increasingly relevant due to its wide range of applications, these are some examples:

- **User Engagement:** emotion detection identifies confusion, boredom, and anger to

improve features and recreate interest. It also reveals happiness with entertaining content.

- **Content Moderation:** detecting anger, and disgust facilitates flagging abusive posts and disturbing content like violence for moderation.
- **Marketing:** interest, joy and surprise reactions inform creative strategy and targeting for personalized, viral-ready campaigns. Anger and sadness guide message adjustments.

8.2 X Platform

With over 500 million tweets posted daily, X provides a valuable platform for detecting emotional states [38][66] based on textual and visual signals. Researchers have developed techniques to automatically identify emotions [84] like happiness, sadness, anger, and fear from the language, emojis, and emoticons used in tweets. Overall, X's vast public emotion data offers a prolific resource to keep improving automated emotion recognition technologies. With growing computational linguistics capabilities, fine-grained analysis of emotive expressions on X shows promise for many downstream applications from mental health monitoring to stock prediction and consumer insights to political analytics.

Additionally, emotion detection was very effective during COVID-19 lockdowns [63]. Analysis of used social media words revealed increased fear, sadness, and loneliness in isolated populations during the pandemic. This demonstrates the technique's ability to unobtrusively monitor mental health through expressed emotions.

9 Challenges in emotion detection

Emotion detection from text data is full of challenges for several reasons precisely in the virtual world. The era of the Internet has led to the generation of a vast amount of informal text data, which presents various challenges for emotion analysis. Social networking sites, in particular, pose challenges such as spelling mistakes, new slang (dialectal language), and incorrect grammar usage, making it difficult for machines to perform emotion analysis.

Additionally, individuals may not express their emotions clearly, further complicating the process of emotion detection from real-world data.

The lack of resources is a problem for emotion recognition because some statistical techniques need a sizable annotated dataset. While collecting the data is not hard, manually classifying the massive dataset takes a lot of effort and is not as accurate. The complexity and subtlety of emotions, the dearth of labelled datasets, and the requirement for domain-specific knowledge are some of the factors contributing to this difficulty.[69].

The Web slang is another prevalent issue that frequently appears in postings and conversations on Instagram [43] , X, and Facebook. His expanding language is an enormous challenge to current lexicons and trained models .[69].

10 Conclusion

Emotion detection is an emerging interdisciplinary field with applications in mental healthcare, marketing, and human-computer interaction. Detecting emotions involves analysing facial expressions, speech, text and physiological signals, but it remains complex due to challenges like contextual understanding and cultural differences. Despite advancements, emotion detection systems need to become more accurate, nuanced and user responsive. When carefully designed, these systems have the potential to significantly improve well-being, communication, and decision-making in both personal and professional contexts. Creating efficient systems needs robust techniques.

The next section introduces techniques used in the context of emotion detection mainly the machine learning techniques.

CHAPTER 2

Machine learning

1 Introduction

Artificial intelligence (AI) refers to the capability of machines to exhibit human-like intelligence and perform complex tasks like sensing, learning, reasoning and decision making. Machine learning is a subset of AI that enables algorithms and systems to automatically learn from data and improve their performance over time without explicit programming. In this chapter, we will provide an overview of artificial intelligence especially machine learning approaches focusing on different types of machine learning algorithms.

2 Artificial intelligence

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions [45]. This can include learning, reasoning, problem-solving, perception, and language understanding. AI systems are designed to perform tasks that would typically require human intelligence, ranging from simple ones like recognizing speech or images, to more complex ones like decision-making and translating languages.

3 Machine Learning

Machine learning (ML) is a subset of AI that enables algorithms and systems to automatically learn from data and improve their performance over time without explicit programming. In 1959, **Arthur Samuel** described ML as the “*field of study that gives computers the ability to learn without being explicitly programmed*”. According to **Tom M. Mitchell**’s definition of ML “*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience reinforcement learning E* ”[65].

4 Types of Machine Learning

Several broad categories of machine learning differ in terms of how they operate and what kind of data they require for training:

4.1 Supervised learning

Supervised Learning [67] involves training the model on labeled data and testing it on unlabeled data. Its fundamental architecture begins with dataset collection, the dataset is then partitioned into testing and training data, and then, the data is pre-processed. Extracted features are fed into an algorithm and the model is then trained to learn the features associated with each label. Finally, the model is supplied with the test data and it makes predictions on them by providing the expected labels, as illustrated in Figure 2.1.

Supervised learning tasks are divided into two categories : classification and regression, both are discussed in detail in the following sections.

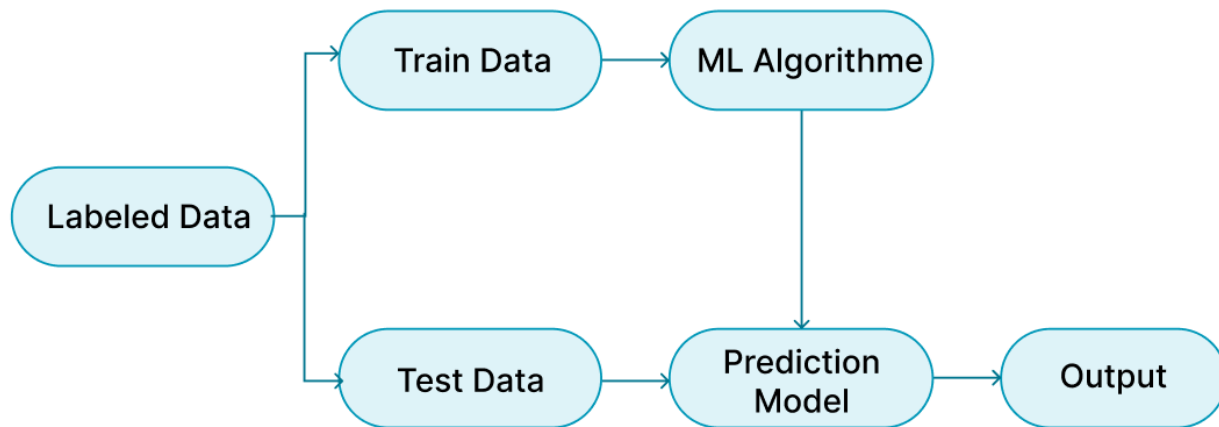


Figure 2.1: Basic Architecture of Supervised Learning [27]

4.1.1 Classification

Classification [27] is a type of supervised machine learning algorithm used to predict discrete categorical labels. It learns from training data how to assign observations to different class labels or categories. Classification models classify inputs into discrete outputs based on learned patterns. Common algorithms include logistic regression, decision trees, random forests, naive Bayes and support vector machines. Classification is useful for tasks like image recognition, spam detection, medical diagnosis and customer churn prediction.

4.1.2 Regression

Regression [27] is a type of supervised machine learning algorithm used to predict continuous target variables. It learns the mapping function between input and output variables from labeled training data. Regression models estimate relationships between features and the target by fitting curves/lines to minimize the prediction error. Common regression algorithms include linear regression, logistic regression, polynomial regression and regression trees. Regression is useful for forecasting, estimation, and predictive modeling tasks with continuous numeric outputs.

4.2 Reinforcement Learning

Reinforcement learning [25][11] is commonly used in robotics, gaming, and navigation applications. It involves an algorithm learning in a dynamic environment to achieve a goal without explicit instructions. Through trial-and-error interactions, reinforcement learning determines which actions produce the greatest rewards. For example, a chess-playing reinforcement learning algorithm would progressively learn the game by playing against opponents and experimenting with actions to win.

Reinforcement learning has three key components: the learning agent, the environment it interacts with, and the actions it can take. The objective is for the agent to choose actions that maximize cumulative reward over time. Following an optimal policy allows the agent to reach its goal much faster. Therefore, reinforcement learning aims to learn the best policy through experience and reward/penalty feedback.

4.3 Unsupervised Learning

Unsupervised learning [62][102] algorithms learn from unlabeled training data without explicit correct answers or supervision. They uncover hidden patterns and relationships in the data to learn more about its structure. The similar data is grouped in clusters. Common unsupervised learning techniques include clustering algorithms like k-means which group data points with similar traits and dimensionality reduction techniques like principal component analysis which find the most salient features. Some major types of unsupervised learning algorithms are:

4.3.1 Dimensionality Reduction Algorithms

Dimensionality reduction techniques [25] are an effective solution for the problem of the curse of dimensionality. As the number of dimensions or features increases, the volume of the data space grows exponentially. This causes the available data to become very sparse. This sparsity poses an issue for methods that require statistical power, since the amount of data needed grows rapidly with more dimensions.

Dimensionality reduction involves methods for decreasing the number of dimensions characterizing an object. The main goals are to eliminate irrelevant and redundant data,

reduce computational costs and improve data quality and organization. Similar to clustering approaches, dimensionality reduction methods seek to exploit the inherent structure within data in an unsupervised fashion. Many techniques can also be adapted for classification and regression tasks.

Some common dimensionality reduction algorithms include: Principal Component Analysis (PCA), Principal Component Regression (PCR) and Partial Least Squares Regression (PLSR)... etc

4.3.2 Clustering

Clustering is a key technique in unsupervised learning that groups unlabeled data points based on similarity. The goal of clustering is to partition data into distinct groups called clusters, where objects within a cluster are more similar to each other than objects in other clusters.

Clustering reduces data complexity by grouping many granular points into a smaller number of clusters. This compression leads to some loss of finer distinctions between data points but makes the data much simpler to analyze at a broader level.

For instance, Clustering algorithms can analyze text data to identify and group posts that express similar emotions in social media, aiding in understanding public sentiment or trends.

Overall, clustering represents a crucial unsupervised technique for identifying groups and patterns within datasets in the absence of predefined training labels. This method finds extensive application in domains such as customer segmentation, image segmentation, social network analysis, among others.

4.3.2.1 General Techniques of Clusters

Different sorts of clustering techniques are available to handle various kinds of unique data.

- a) **Center-Based Clusters:** in center based technique [79], the cluster is a collection of objects where each object is closer or more similar to the center of its own cluster than to the center of any other cluster. This center, often referred to as the centroid, is typically the average of all the points in the cluster. Alternatively, the center can be a medoid, which is the most representative point of the cluster. The concept of a centroid or a medoid helps to define the core of the cluster, ensuring that the objects within a cluster are more closely or similarly associated with their own cluster's center than with the centers of other clusters.
- b) **Well-Separated Clusters:** in this case [51], a cluster is defined as a set of nodes or points, where each node or point within the cluster is closer or more similar to every other node or point in the same cluster than to any node or point outside of it. This proximity or similarity ensures that if the clusters are sufficiently well-separated, various clustering methods can effectively distinguish and group them. Essentially, the cohesiveness within a cluster and the distinction between different clusters are key factors that contribute to the effectiveness of clustering techniques.
- c) **Density-Based Clusters:** the technique [79][51] is based on the principal that a cluster is a dense region of points, distinctly separated by low-density areas from other high-density regions. This concept is particularly useful in scenarios where clusters are intertwined or irregular, and where there are factors like noise and outliers. The delineation of a cluster is based on the density of points, ensuring that each cluster is a compact group separated from other clusters by areas of lower point density. This approach is effective in managing complex data structures where traditional clustering methods might struggle due to the presence of irregularities and external noise factors.
- d) **Contiguous clusters:** in Contiguous clusters technique [51], the cluster is a collection of points where each point within the cluster is nearer or more similar to one or more other points in the same cluster compared to any point outside of it. This proximity or similarity between points within a cluster emphasizes the cohesiveness of the cluster, ensuring that each point shares a stronger connection with points within the same cluster than with those outside it. This concept is fundamental in identifying and grouping points into distinct clusters based on their relative closeness or similarity.

4.3.2.2 Methods of Clustering

Traditional clustering methods are generally classified into hierarchical, partitioning, and density-based techniques. However, categorizing these clustering approaches is not simple or universally agreed upon. In practice, there is often an overlap among these groups.

a) Hierarchical Methods: hierarchical [12][57] clustering creates a multi-level breakdown of a dataset or group of objects based on specific criteria. This process is represented through a dendrogram, resembling a tree diagram, which tracks the history of merging or dividing sequences. By cutting the dendrogram at an appropriate level, any number of clusters can be derived. Each cluster node encompasses smaller, child clusters, and clusters at the same level divide the data points contained in their shared parent cluster. This method facilitates the examination of data at various levels of detail. Hierarchical clustering is divided into two types: agglomerative (bottom-up) and divisive (top-down). In agglomerative clustering, the process begins with clusters consisting of a single point each and progressively merges clusters that are most similar to each other. On the other hand, divisive clustering starts with all data points in a single cluster and iteratively divides the most suitable cluster. This procedure continues until a predetermined stopping point is reached, often determined by the desired number of clusters, denoted as 'k'.

b) Partitioning Methods: partitioning methods [57] typically create M distinct clusters, with each object assigned to one cluster. Each cluster can be represented by a centroid or a representative, which serves as a collective summary of all objects in the cluster. The nature of this summary varies depending on the type of objects being clustered. For instance, when clustering real-valued data, the average of the attribute vectors of all objects in a cluster can be used as an effective representative. However, different types of centroids might be necessary for other scenarios, such as using a list of common keywords to represent a cluster of documents, where the keywords are chosen based on their frequency across a minimum number of documents in the cluster. In cases where there are numerous clusters, these centroids themselves can be clustered to form a hierarchical structure within the dataset. There are many methods and algorithms of partitioning clustering, among which we can mention K-means Algorithm and Medoids Algorithm. .

K-means Algorithm: this algorithm [97][14] aims to identify K divisions that meet a specific criterion. It starts by selecting some dots as the initial focal points for the clusters (commonly, the first K sample dots are used for this purpose). Next, the rest of the sample dots are grouped with these focal points based on the principle of minimum distance. This step leads to an initial classification. If this classification appears to be unsatisfactory, it is adjusted by recalculating the focal points of each cluster. This process is repeated multiple times until a satisfactory classification is achieved.

Medoids Algorithm: k-medoids identify the most representative data point within a cluster. Using k-medoids for representation offers two main benefits. Firstly, it is versatile, imposing no restrictions on the types of attributes used. Secondly, the selection of medoids is influenced by the location of the majority of points in a cluster, making it less vulnerable to outliers. Clusters are formed around the chosen medoids, grouping points that are nearest to them, and the goal is to minimize an objective function, typically the average distance or another measure of dissimilarity between a point and its nearest medoid.

c) Density-Based Algorithms: density-based [79][95] algorithms are a class of clustering methods that identify clusters based on the density of data points in a region. Unlike other clustering techniques that require pre-specifying the number of clusters, density-based methods are particularly adept at discovering clusters of arbitrary shapes and sizes, and they are inherently robust against outliers.

The core principle of these algorithms is to consider areas with a high density of data points as a single cluster while treating regions with low density as noise or outliers. This approach allows for more flexibility and adaptability in various data environments.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is the most widely used density-based clustering algorithm.

DBSCAN algorithm: this is an appropriate algorithm [18][93] for finding outliers 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.

d) **FUZZY C-Means Clustering:** FUZZY [53] C-Means clustering method assigns each data point to a cluster with a certain membership level. The technique initiates by making an initial estimation of the cluster centers, which symbolize the average location of each cluster.

4.3.2.3 Comparison between algorithms of clustering

Clustering algorithms are pivotal in organizing data into meaningful groups, each characterized by similarities among its members. This comparison outlines the strengths and limitations of prominent clustering techniques. Understanding these algorithmic nuances facilitates informed decisions in selecting the most suitable approach for data analysis tasks.

Clustering Algorithm	Advantages	Disadvantages
K-means	<ul style="list-style-type: none"> • It's relatively easy to scale and straightforward. • Appropriate for datasets featuring compact, well-separated spherical clusters. 	<ul style="list-style-type: none"> • Limited ability to provide accurate descriptions of clusters. • Requires users to predefine the number of clusters.
DBSCAN	<ul style="list-style-type: none"> • It identifies clusters with varied shapes, regardless of their form. • It effectively manages noise and outliers. 	<ul style="list-style-type: none"> • Sensitivity to the configuration of input parameters is notable. • Inadequate representations of clusters are evident.

Agglomerative**Hierarchical**

- Eliminates the requirement to specify the number of clusters beforehand.
- Generates a complete hierarchy of clustered data.
- High time and space complexity
- Incapable of adjustments after splitting or merging decisions are finalized.

Fuzzy**Clustering**

- Performs more effectively than the conventional hard clustering method, such as the k-means algorithm, particularly in scenarios involving overlapping data points.
- Data points are not limited to membership in a single cluster; instead, they can possess fractional memberships across multiple clusters.
- Relatively slower in execution due to the necessity of calculating the membership of every data point across each cluster.
- Prone to being influenced by the initial setup of the weight matrix.

In summary, each algorithm has advantages and disadvantages. K-means is simple and fast but makes assumptions. DBSCAN is flexible but does not handle densities well. Hierarchical clustering provides visualization but has high complexity. Fuzzy clustering allows partial membership but results depend on initialization.

5 Conclusion

In this chapter, we provided a brief overview of AI and ML. It covers the definition of different related concepts especially supervised, unsupervised and reinforcement learning. Key algorithms like k-means and DBSCAN are presented as well, emphasizing the impor-

tance of choosing the right method for specific tasks. The chapter also accentuate different clustering techniques precisely ensembles clustering techniques.

The next section provides a detailed description of an amount rang of clustering techniques used in the text analysis field specially in emotion detection.

CHAPTER 3

Text Clustering

1 Introduction

This chapter delves into the critical area of text clustering, focusing on its common applications to short texts in social media. It begins by defining text clustering and distinguishing between long and short text clustering. The chapter then explores the practical applications of short text clustering and the challenges it presents, such as feature sparsity and high dimensionality. A comparative study of various text clustering techniques for emotion detection in social media data is presented, highlighting the effectiveness of ensemble learning methods. The chapter also discusses the processes, advantages, applications and challenges of ensemble clustering also future directions, concluding with an overview of related works.

2 Definition of Text Clustering

Text clustering [48], also known as text categorization or document clustering, is a natural language processing (NLP) technique. Can be conceptually explained as the process of dividing text data into clusters based on their similarity, where each cluster comprises data points in a multi-dimensional space that are similar to one another. Text clustering groups texts based on shared properties, such as similar words, subjects, or themes. Clusters can be formed based on various features extracted from the text, including word frequencies, semantic similarity, or topic distributions, Algorithms for text clustering organize

documents into clusters by optimizing similarities within each cluster while minimizing similarities between clusters. There are two subtypes or domains within the broader field of text clustering, such as **short text clustering** and **long text clustering**.

2.1 Long Text Clustering

Long text clustering [19] involves grouping extensive textual materials, such as articles, academic papers, or lengthy books, which often cover diverse and intricate subject matters. This form of clustering poses computational scalability issues because conventional clustering methods may struggle to handle the size and complexity of such datasets, leading to inefficiencies or impracticalities in processing them.

2.2 Short Text Clustering

Clustering short texts [7][105] are one of the most important text analysis methods to help extract knowledge from the Internet, including on social media, in product descriptions, in advertisement text, on questions and answers websites and in many other applications. Short texts might be difficult to find knowledge in since they are characterized by a lack of context. Short texts can be found in a variety of settings, including product descriptions, chat messages, tweets, search queries, and online reviews.

Short texts pose a clustering challenge due to their disorderly characteristics, often including noise, slang, emojis, misspellings, abbreviations, and grammatical mistakes. Tweets serve as a prime example of such difficulties.

3 Applications of Short Text Clustering

Various clustering techniques have found applications across multiple real-world domains. The following disciplines and sectors make use of clustering methods:

- **In information retrieval (IR):** clustering techniques have been applied in diverse scenarios, including the clustering of large datasets. Within search engines, text clustering is pivotal for enhancing document retrieval efficacy by categorizing and indexing relevant documents.

- **Internet of Things (IoT):** a number of areas have made IoT their primary focus because to the quick growth of technology. A global positioning system, radio frequency identification technology, sensors, and other IoT devices are some of the tools used in data collection in the Internet of Things. Distributed clustering, which is necessary for wireless sensor networks, is accomplished through the use of clustering algorithms.
- **Emotion Recognition (ER):** text clustering can group documents or sentences by sentiment. This can be used to analyse sentiment and emotion in data like reviews, social media, etc.
- **Summarization:** clustering sentences/phrases of a document using semantic similarity can help generate key phrase or extractive summaries.

4 Challenges in Short Texts Clustering

Text clustering researchers are confronted with a number of challenges. Compared to broad text clustering, short text clustering has a distinct issue.[\[96\]](#) The most frequent difficulties with short text clustering will be covered in the next section.

4.1 Sparse Feature Vector

In document or large text clustering algorithms, each document is represented by a feature vector. Numerical values for the features that match document phrases are contained in this vector. Short texts include extremely few words, hence the feature vector that is created from them is typically sparse in nature. One of the main problems in clustering short text data is the sparsity of the feature vector, which is hard to solve.

4.2 Polysemy

The existence of various interpretations for a single word presents a substantial hurdle in text clustering. For example, the term "table" might denote furniture, a data structure, or mathematical tables. Assigning the correct category for such terms becomes intricate. Unlike longer documents where context aids in clarifying the meaning, short texts lack the requisite context due to their brevity. Consequently, grasping the context of a word in short text becomes particularly arduous, given the sparse nature of the text.

4.3 Synonymy

Identifying words with identical meanings presents a challenge in text clustering. For instance, words like "Beautiful," "Attractive," "Pretty," "Lovely," and "Stunning" share the same meaning. Deciding which cluster to assign such words becomes particularly difficult, especially when they appear in short texts.

5 Text Clustering and Emotion Detection

Integrating text clustering with emotion detection [107] allows for identifying clusters of text documents based on both semantic content and emotional tone. This integration provides deeper insights into the emotional context within text data, enabling nuanced emotion analysis at scale. Text clustering organizes documents by thematic content, while emotion detection identifies predominant emotions within each cluster. This combined analysis is valuable for social media analysis, customer feedback mining, and product review opinion mining, where understanding emotional content is critical for decision-making and user engagement.

6 A Comparative Study of Text Clustering Techniques for Emotion Detection in Social Media Data

In the modern era of technology, a significant portion of the global population utilizes the Internet to communicate through various mediums including text, images, audio, and video. Individuals from diverse backgrounds engage in exchanging information and expressing their perspectives on current events through social media platforms. There is a necessity to comprehend and interpret the impact of such extensive textual information on individuals by analyzing their emotional responses. We have collected some previous works that mention some of the techniques used in this field.

Table 3.1: Summary of Previous works for Emotion Recognition from Text

Ref no.	Main Focus	Approach	Key Contributions	Results
[107]	Addressing sparse features in emotion recognition from short texts	- Representing short texts with word cluster features - Introducing a novel word clustering algorithm - Employing a unique feature weighting scheme	- Proposed methods to enhance emotion recognition performance - Conducted emotion classification experiments using various features and weighting schemes - Word cluster features and the proposed weighting scheme partly resolved issues with feature sparseness and emotion recognition performance	Experimental results suggest that the word cluster features and the proposed weighting scheme partly resolved problems with feature sparseness and emotion recognition performance
[90]	Extracting emotional insights from Twitter data	- Transforming Twitter data into a vector of eight distinct emotions - Utilizing supervised machine learning techniques such as K-means, Naive Bayes and SVM for emotion classification	- Focused on extracting emotional insights from Twitter data - Employed supervised learning methods for emotion classification - Transformed Twitter data into a vector representing eight distinct emotions	

[6]	Unsupervised emotion detection at the sentence level	<ul style="list-style-type: none"> - Innovative unsupervised approach not reliant on pre-existing lexicons - Calculation of emotion vectors for potential affect-bearing words based on semantic connections and syntactic dependencies within sentence structures 	<ul style="list-style-type: none"> - Introduction of a versatile unsupervised approach for emotion detection - Calculation of emotion vectors based on semantic and syntactic features - Framework demonstrated superior effectiveness compared to recent unsupervised methods 	Extensive evaluations on diverse datasets demonstrated the framework's superior effectiveness compared to recent unsupervised methods
[80]	Understanding users' feelings towards specific topics	- Text-based emotion recognition approach	<ul style="list-style-type: none"> - Proposal of a method using personal text data for emotion recognition - Application of Dominant Meaning Technique for emotion recognition 	Promising experimental results reported on the tested dataset based on the proposed algorithm. It outperforms other methods in precision, recall, and F-measure

7 Ensemble Learning

Ensemble learning frameworks [89] is a machine learning paradigm where multiple models (such as classifiers or experts) are strategically generated and combined to solve a particular computational intelligence problem. An ensemble is constructed with a combination of multiple base learning algorithms to achieve better predictive performance compared to any single learning algorithm or model.

The key idea is to train multiple base models and then combine their predictions into a single final prediction to improve robustness and generalizability over a single model. This is typically done for supervised learning tasks like classification and regression.

Ensemble methods can also be applied to unsupervised learning tasks like clustering, which is known as ensemble clustering. In ensemble clustering, multiple base clustering results are generated and then combined to obtain a consensus clustering solution.

8 Methods of Ensemble supervised Learning

Ensemble methods are categorized into two primary frameworks: the dependent framework and the independent framework. Within the dependent framework, the outcome of each inducer influences the development of the subsequent inducer. Here, insights gained from previous iterations inform the learning process in subsequent ones. Conversely, in the independent framework, each inducer is constructed autonomously, without reliance on other inducers. these are the most popular ensemble methods of both frameworks[89]:

- **AdaBoost:** [35] is the most well-known dependent algorithm for building an ensemble model. The main idea of AdaBoost is to focus on instances that were previously misclassified when training a new inducer.
- **Bagging:** [15] is a simple yet effective approach for generating an ensemble of independent models in which each inducer is trained using a sample of instances taken from the original dataset as a replacement. In order to ensure a sufficient amount of instances per inducer, each sample usually contains the same number of instances as in the original dataset. Majority voting of the inducers' predictions is performed to determine the final prediction of an unseen instance.
- **Random forest and random subspace methods:** Random forest's popularity continues to increase, primarily due to its simplicity and predictive performance. In addition, random forest is considered an easy to tune method compared to other methods that require careful tuning. The random forest algorithm was originally developed for using decision trees as base learners, largely because of the process of choosing different feature subsets when splitting the nodes. However, this step can easily be

replaced by using the broader random subspace method (RSM; Ho, 1998) which can be applied with other types of inducers.

- **Gradient boosting machines (GBM):**In GBM [36] the training of each inducer is dependent on inducers that have already been trained. The main difference between GBM and other techniques is that in GBM optimization is applied in the function space. It includes a learning procedure in which the goal is to construct the base learners so that they are maximally correlated with the negative gradient of the loss function, associated with the whole ensemble. More specifically, in GBM a sequence of regression trees is computed, where each successive tree predicts the pseudo-residuals of the preceding trees given an arbitrary differentiable loss function.
- **Rotation forest:** [85] is a method that generates diversity among decision tree inducers by training each inducer on the whole dataset in a rotated feature space.

9 Ensemble Clustering

Clustering ensemble also named consensus clustering [1][5][16] is a clustering framework that combines multiple clustering results to obtain a consolidated clustering that is better than any of the individual clustering. It has emerged as an effective way to improve robustness, stability and accuracy of unsupervised classification solutions.

The main idea is to run multiple clustering algorithms on the same dataset using different initializations or parameters. Each clustering algorithm may produce different clustering structures due to differences in biases and sensitivities to parameters and initialization. Instead of selecting the best clustering or using model averaging, the results from multiple runs are integrated to find areas of agreement between clusterings. This consensus across multiple partitions is then used to generate a final consolidated clustering.

Overall, clustering ensembles provide a principled way to build robust unsupervised learning systems by combining results across multiple clustering solutions.

10 Process of ensemble clustering

Every clustering ensemble method is made up of two steps: process of generating individual partitions and combining them to generate the final partition (Consensus function),

as shown in Figure 3.1

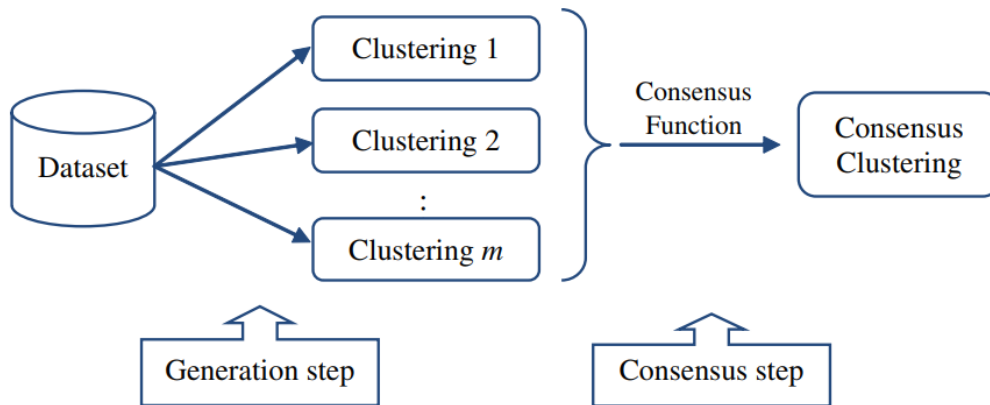


Figure 3.1: Diagram of the general process of cluster ensemble [99]

10.1 Generation Process

The generation step[1][16][99] is the process of creating multiple partitions (M partitions) of the given dataset that will serve as input to the cluster ensemble method. The key goals during generation are to produce partitions that are diverse from each other but also individually of high quality.

There are no specific constraints imposed on how these partitions must be obtained, giving flexibility to generate diversity through different techniques. For example, partitions can be created using different clustering algorithms like k -means, hierarchical clustering, DBSCAN, etc. The same algorithm can also be run multiple times with different parameter initializations or on different random subsamples of the data. Additionally, diversity can be introduced by transforming the data in different ways before clustering, like using different feature subsets, applying dimensionality reductions, or projecting the data into lower dimensional subspaces.

Fundamentally, the generation step aims to create a diverse set of high-quality partitions. The fewer constraints, the more flexibility to produce partitions capturing different perspectives of the clustering structure in the data. This diversity allows the ensemble to

achieve better performance than individual partitions. The high-quality partitions ensure meaningful information is combined. By bringing together diverse, high-quality partitions, the ensemble can find commonalities and inconsistencies that improve the unified clustering.

There are different techniques [16][1] used for generating the individual clusters :

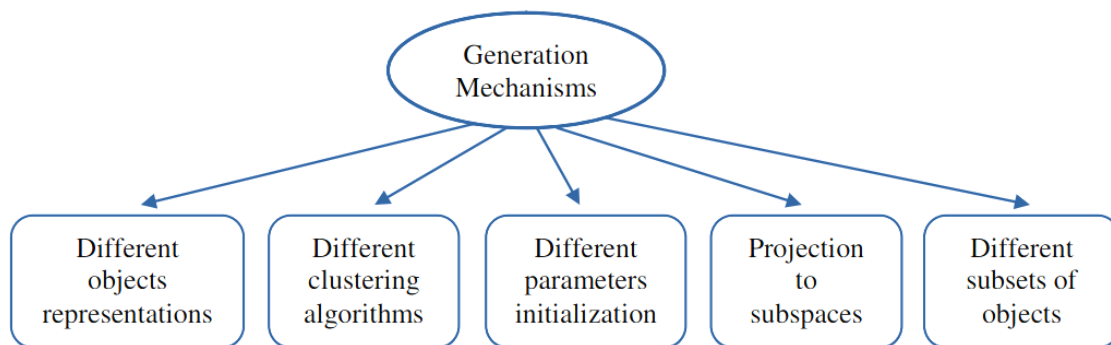


Figure 3.2: Diagram of the principal clustering ensemble generation mechanisms [99]

10.1.1 Different object representations

Diversity is obtained by generating partitions using different subsets of attributes, changing parameter forms during generation, and collecting different information per object, enabling complementary perspectives like in ensemble methods.

10.1.2 Different clustering algorithms

Have unique biases and can produce varied partitions on the same data; generating an ensemble of partitions using diverse algorithms takes advantage of their different perspectives.

10.1.3 Different parameter initialization

By varying initialization parameters that directly or indirectly affect cluster number and starting points, clustering algorithms can produce diverse partitions on the same data by optimizing different local objectives.

10.1.4 Subspace projection

Using dimensionality reduction techniques like random projections or random attribute subsets creates different clusterings from varied perspectives by representing patterns with different attribute subsets.

10.1.5 Subsets of objects

By clustering different random subsamples of the data, like bootstrapping in supervised methods, diverse partitions can be generated from the varied perspectives of different subsets of examples.

10.2 Consensus Process

Consensus process [5][16][8] is the process of combining multiple clusterings or partitions of the same dataset, produced by different algorithms or components, in order to obtain a single consolidated and robust clustering or partition. It utilizes the assignment information of examples to clusters from the individual partitions to determine the final consensuated partition. The goal is to leverage the different biases and perspectives of the individual partitions to arrive at an improved overall clustering. A consensus function is used to aggregate the individual partitions and evaluate the quality of the final solution. Overall, consensus process aims to boost robustness, stability, and accuracy by synthesizing the outputs of diverse clustering approaches on the same data.

There are two main consensus function approaches:

10.2.1 Co-occurrence based approach

“It firstly computes the co-occurrence of objects in the members and then determines their cluster labels to produce a consensus result. Simply, it counts the occurrence of an object in one cluster, or the occurrence of a pair of objects in the same cluster, and generates the final clustering result by a voting process among the objects”. [8]

10.2.2 The median partition approach

This treats the consensus function as an optimisation problem of finding the median partition with respect to the cluster ensemble. The median partition is defined as “the

partition that maximises the similarity with all partitions in the clustering ensemble”[8].

11 Properties and Advantages of using ensemble methods over single clustering algorithms

Some research [98][34][16] has tried to define properties that support using ensemble clustering methods, but there is no agreement on what those key properties are. This remains an open question. Defining these properties is difficult because most proposed properties are hard to prove. Four relevant properties are often cited:

- **Robustness:** The combination process must have better average performance than the single clustering algorithms.
- **Consistency:** The combination result should be very similar to the individual clustering algorithm results.
- **Novelty:** The ensemble clustering methods should be able to find solutions that individual clustering algorithms cannot reach.
- **Stability:** The combination result should be less sensitive to noise and outliers than the individual clustering results.

Consensus clustering also adds some advantages over classical clustering algorithms

- **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.[16]
- **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.[16]
- **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.[16]

12 Applications of Ensemble Clustering in Text Analysis

Ensemble clustering techniques combine multiple clustering models to improve performance on text analysis tasks such as:

- **Text classification:** Grouping texts into predefined categories or classes. Ensemble clustering can help improve accuracy of automated text classification by combining multiple base cluster models.
- **Document clustering:** Unsupervised grouping of documents into clusters based on their contents and topics. Ensemble clustering generates more robust document clusters than individual algorithms.
- **Sentiment analysis:** Identifying and extracting subjective information like opinions, emotions, and attitudes in text. Ensemble clustering can better capture nuances in sentiment by merging different clustering views.
- **Emotion detection:** Ensemble clustering creates superior emotion classifiers by consolidating diverse individual models for the detection of emotional states like joy, sadness, anger, etc expressed in text.

13 Challenges and Future Directions in Ensemble Clustering

While ensemble clustering has shown promising results in many applications, there remain some key challenges and opportunities for future work. Some of the main challenges include developing efficient methods to generate diverse base clusterings, effectively combining multiple partitions, handling noise and outliers, and scaling ensemble techniques to large datasets. Additional research is needed to determine optimal ensemble configurations for different problem domains. Future directions for ensemble clustering could include integrating with deep neural networks, active semi-supervised learning, and multi-view domain adaptation. There is also scope for advances in ensemble clustering of temporal data, multi-model data, and applications in emerging fields like robotics. Tackling these challenges and innovations will help realize the full potential of ensemble clustering.

14 Emotion Detection Related Works

Detecting emotions across different modalities, such as face, speech, and text plays a crucial role in understanding human behaviour and enhancing human-computer interaction. In this section, we'll delve into a collection of previous works that employ supervised and unsupervised learning techniques to detect emotions through facial expressions, speech patterns, and textual content.

14.1 Supervised learning techniques

14.1.1 Text

- The paper [74], presents a sentiment analysis system employing an ensemble of classifiers to automatically detect emotions in text. The ensemble integrates statistical and knowledge-based methods, comprising two statistical classifiers (Naïve Bayes and Maximum Entropy) and a knowledge-based tool that analyses text structure using a keyword-based approach. Evaluation on various text types, such as news headlines and social media posts, demonstrates satisfactory performance in emotion recognition and polarity identification.
- The study in [101] compares three popular ensemble methods (Bagging, Boosting, and Random Subspace) using five base learners (Naive Bayes, Maximum Entropy, Decision Tree, K Nearest Neighbor, and Support Vector Machine) for sentiment classification. Evaluating ten public sentiment analysis datasets, the study conducts 1200 experiments to assess ensemble learning effectiveness. Results demonstrate significant performance enhancement over individual base learners, with Random Subspace showing superior performance. The findings underscore ensemble learning's viability for sentiment classification.
- The authors of the article [58] proposes an AI approach for automated emotion detection, enhancing machine capabilities like chatbots to adapt communication based on emotional cues. Despite challenges in full automation, machine learning techniques using conversational text data show promise. Experiments utilized lexicon-based and classic ML methods (e.g., Naïve Bayes, SVM) and deep learning with neural networks to build emotion detection models. Neural networks performed well, especially in

detecting sadness. The top model was integrated into a web app and chatbot, improving human-machine interaction. While showing potential, complete automation of emotion detection remains a question, requiring further refinement. The paper also explores philosophical and psychological dimensions of automated emotion detection.

14.1.2 Facial

- The study in [20] illustrates a multi-block deep convolutional neural networks (DCNN) model which was conceived and implemented to identify facial emotions from virtual, stylized, and human characters.
- The work in [37] presents a novel Facial expression recognition (FER) framework using a convolutional neural network (CNN) and soft label that associates multiple emotions with each expression. The results indicate this method achieves competitive or even better performance (FER-2013: 73.73%, SFEW: 55.73%, RAF: 86.31%) compared to state-of-the-art methods.
- The work in the paper [33] introduces a novel Multi-Region Ensemble CNN (MRE-CNN) framework designed for facial expression recognition. The framework enhances the learning capacity of CNN models by capturing both global and local features from multiple sub-regions of the human face. It aggregates weighted prediction scores from each sub-network to generate highly accurate final predictions. Additionally, the study investigates the impact of different face sub-regions on facial expression recognition. Evaluation on two widely-used facial expression databases, AFEW 7.0 [26] and RAF-DB [56], demonstrates that the proposed method achieves state-of-the-art recognition accuracy.

14.1.3 Speech

- This research [44], focuses on recognizing various emotions from audio speech by extracting features and utilizing them for emotion classification. Mel Frequency Cepstral Coefficients (MFCCs) are employed for feature extraction. Six supervised classifiers, including multilayer perceptron (MLP), Random Forest (RF), AdaBoost, support

vector machine (SVM), Gradient Boosting (GB), and Hist Gradient Boosting (HGB), are utilized for classification. A comparative analysis among the classifiers suggests that Ensemble Method emerges as one of the most effective techniques for emotion recognition from speech.

- Po-Yuan Shih in [94], proposes to apply ensemble learning methods to neural networks to enhance the performance of speech emotion recognition tasks.
- In [109] the authors propose the agglutination of prosodic and spectral features from a group of carefully selected features to realize hybrid acoustic features for improving the task of emotion recognition. Experiments were performed to test the effectiveness of the proposed features extracted from speech files of two public databases and used to train five popular ensemble learning algorithms. Results show that random decision forest ensemble learning of the proposed hybrid acoustic features is highly effective for speech emotion recognition.

14.2 Unsupervised learning techniques

14.2.1 Facial

- For facial expression recognition, the paper [70] uses geometric facial features based on the positional relationships between 2D facial landmark points. Specifically, it defines eight types of geometric features calculated from the distances, angles and areas between different landmark point combinations. For the unsupervised clustering approach, the paper adopts the ensemble clustering technique called cluster-based similarity partitioning algorithm (CSPA) proposed by Strehl and Ghosh. CSPA integrates multiple "weak" clusters into a single set of "strong" discriminative clusters. The k-means algorithm is used to generate the weak clusters based on the different facial feature types.
- The paper [106] proposes a novel hybrid sampling-based clustering ensemble approach that combines the strengths of boosting and bagging techniques. Boosting is known to perform well on noise-free data with complex class structures, while bagging is more robust to noisy data. The proposed method extends both boosting and bagging to clustering tasks. It generates input partitions iteratively through a hybrid process inspired by boosting and bagging. A novel consensus function is introduced

to encode the local and global cluster structure of the input partitions into a single representation. Then, a single clustering algorithm is applied to this representation to obtain the final consolidated consensus partition. The approach has been evaluated on 2D synthetic data, benchmark datasets, and real-world facial recognition data. The results show that the proposed hybrid technique outperforms existing benchmark methods across a variety of clustering tasks, leveraging the advantages of both boosting and bagging.

14.2.2 Video

The authors of the paper [61] proposes a novel method to cluster YouTube videos into six emotion categories: angry, disgust, happy, horror, sad, and surprise. Previous studies have only categorized web videos based on the default categories provided by websites like YouTube. The motivation is to improve video search results by categorizing videos based on the emotions they evoke, which has not been explored before. The approach involves collecting data from YouTube videos, using word embedding techniques to transform the video content into vectors, and then employing a clustering algorithm on these vectors to group the videos into the six emotion categories. A clustering ensemble method is used to obtain the final clustering results. The performance of this proposed method is compared against a state-of-the-art technique, Term Frequency-Inverse Document Frequency (TF-IDF) based on the Vector Space Model, on a benchmark dataset for web video analysis. The results demonstrate that the best performance is achieved by applying the clustering ensemble approach, reflecting the feasibility of clustering web videos into suitable emotion categories.

14.2.3 Text

The proposed approach in the research presented in [64] involves the development and evaluation of an ensemble clustering method for automatic labelling of text data according to the Ekman emotional model, specifically focusing on the Algerian dialect derived from Twitter. The method combines multiple clustering algorithms to produce a single prediction, leveraging a pre-trained BERT model [24] designed for multilingual text representation, including Arabic dialects. By integrating ensemble clustering techniques with BERT, the approach aims to accurately capture the nuanced emotions expressed in Algerian

tweets.

We have thoroughly reviewed the literature for related works on unsupervised approaches, particularly in text analysis, but it appears we have encountered a significant gap. There is a noticeable lack of studies investigating unsupervised methods for emotion detection in text data. Consequently, we propose to address this area with our own research. We aim to develop an unsupervised approach for emotion detection, utilizing ensemble clustering techniques and a pre-trained BERT model for multilingual text.

15 Conclusion

In conclusion, this chapter highlighted significant advancements in text clustering and emotion detection from social media data. It demonstrated that ensemble learning methods, which leverage multiple models, offer superior performance. The analysis of various clustering algorithms emphasized the importance of diversity and consensus in improving accuracy. These findings enhance the field of emotion detection and lay a foundation for future research to refine these techniques further. In the following chapter, we will illustrate the design and implementation of our work within the realm of emotion detection, employing clustering ensemble techniques.

CHAPTER 4

Conception And implementation

1 Introduction

X holds a prominent position among social media platforms, offering a valuable repository of data that attracts the attention of analysts and researchers worldwide. Its brevity of tweets makes it particularly appealing for studies focused on emotion detection and sentiment analysis. In the context of this rich landscape, our initiative aims to extend and enhance the existing work on emotion detection from Algerian dialectical tweets, while also exploring the English-language data processing in the same context.

Our research builds upon the foundation established by previous efforts in extracting emotions from Algerian dialectical tweets. While we acknowledge the significance of this prior work, our focus lies squarely on advancing the methodology, particularly in the domain of ensemble clustering. In the field of ensemble clustering, the generation step emerges as the focal point of our investigation. By refining and enhancing this crucial stage, we aim to unlock deeper insights into the emotional fabric woven within Algerian dialectical tweets. In parallel, we make a special interest to the English-language data processing.

In this chapter, we will illustrate the various steps of the conception and implementation of our work which realise our contribution within the context of emotion detection using clustering ensemble techniques.

2 Conception

Emotion detection from textual data is a multifaceted process that involves several crucial steps, from data collection and preprocessing to the application of advanced machine learning models for accurate emotion.

In Figure 4.1 , we present a visual representation of the sequential steps involved in our methodology, illustrating the flow of data from collection to preprocessing and model application. This diagram serves as a guide to understanding the systematic approach adopted in our study and highlights the integration of various techniques and technologies for effective emotion detection from textual data.

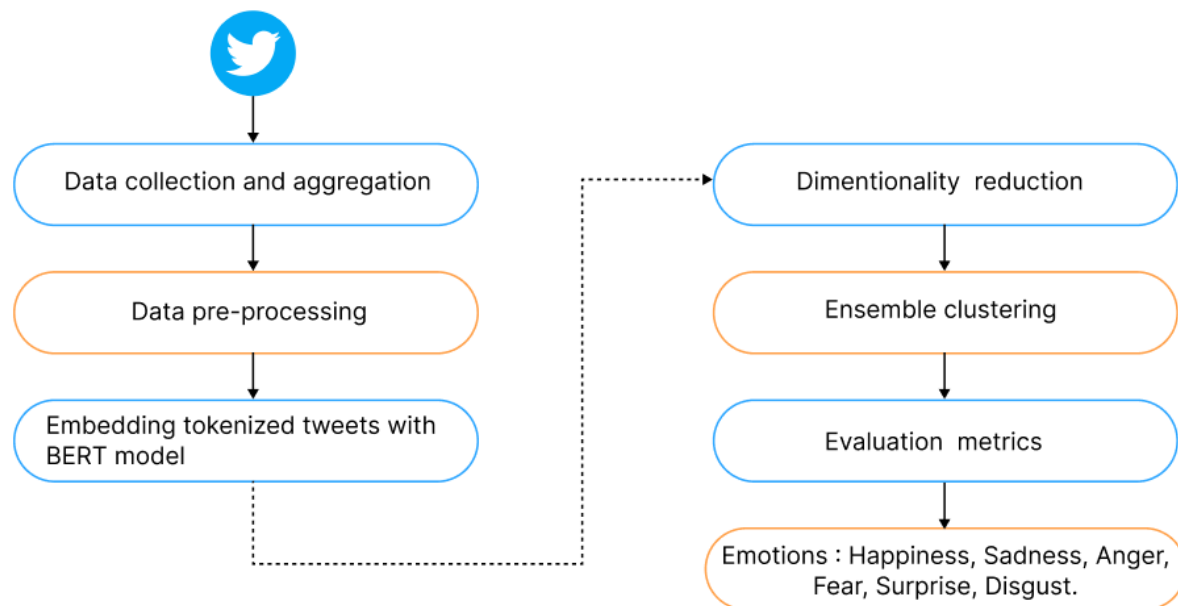


Figure 4.1: Basic steps of Emotion Detection in our work

2.1 Collect data

In recent years, the study of emotion detection from textual data has gained significant traction due to its potential applications in sentiment analysis, and customer feedback analysis, however, gathering diverse and representative datasets for training emotion detection models poses a considerable challenge.

To address this challenge, researchers often turn to various sources that provide databases containing textual data. In our study, we focused on leveraging two prominent sources: the Twitter API and Kaggle. These platforms offer access to vast repositories of text data, encompassing a wide range of topics, and languages,

2.1.1 Twitter API

The Twitter API [103] offers access to X's public data via its Application Programming Interface (API). This interface comprises a series of programmatic endpoints enabling users to engage with X's platform. Through the Twitter API, developers can access a variety of data, including tweets, user profiles, spaces, direct messages, lists, trends, media, and locations.

2.1.2 Kaggle

Kaggle [50] is renowned as the largest data science community globally, providing robust tools and resources tailored to assist individuals in accomplishing their data science objectives. Within Kaggle, users have access to a customizable Jupyter Notebooks environment that requires no setup. Additionally, Kaggle offers the utilization of GPUs at no expense, along with an extensive repository containing data and code contributions from the community.

2.2 Data preprocessing

The preprocessing is a process that comprises several tasks distributed across multiple phases as illustrated in figure 4.2.

1. **Removing Duplicate Letters:** This step helps to normalize words with repeated letters by limiting the repetition to a maximum of two consecutive occurrences.
2. **Text Cleaning:** Cleaning the text is essential for maintaining consistency and enhancing input data quality. Depending on the specific task at hand, this process entails eliminating special characters, punctuation, numbers, URLs, mentions, stock market tickers, and outdated retweet indicators. Additionally, for Arabic text, diacritics, letter elongation, and Arabic question marks are removed to ensure data integrity and uniformity.

3. **Removing Hashtags:** A hashtag consists of letters, numbers, and/or emojis preceded by the # symbol. These hashtags serve to categorize or tag content, enhancing its visibility. Thus, they contain pertinent information useful for our task. Consequently, we simply removed the # symbol and retained the associated words.
4. **Formatting:** This step involves removing new line characters, replacing underscores with spaces, and removing extra spaces.
5. **Lowercasing:** Changing the text to lowercase ensures uniformity and prevents the model from distinguishing between the same word in various cases as distinct entities.
6. **Removing Stop Words:** Stop words are common words in a language that carry little meaningful information, such as articles, prepositions, and conjunctions. These words are typically removed from text data before further processing or analysis, as they can introduce noise and reduce the effectiveness of machine learning models. In our data, we first tokenize the tweet text into individual words using the `word_tokenize` function from the Natural Language Toolkit (NLTK) library. Then, we create a set of Arabic stop words using the `stopwords.words('arabic')` or `stopwords.words('english')` function, which retrieves a list of stop words from the NLTK corpus for the Arabic or English language. Finally, we create a new string by joining the words from the tokenized tweet, excluding those that are present in the set of Arabic or English stop words.

After experimenting with the removal of stop words, we found that it resulted in significantly worse performance. Therefore, we decided not to remove stop words in our preprocessing step.

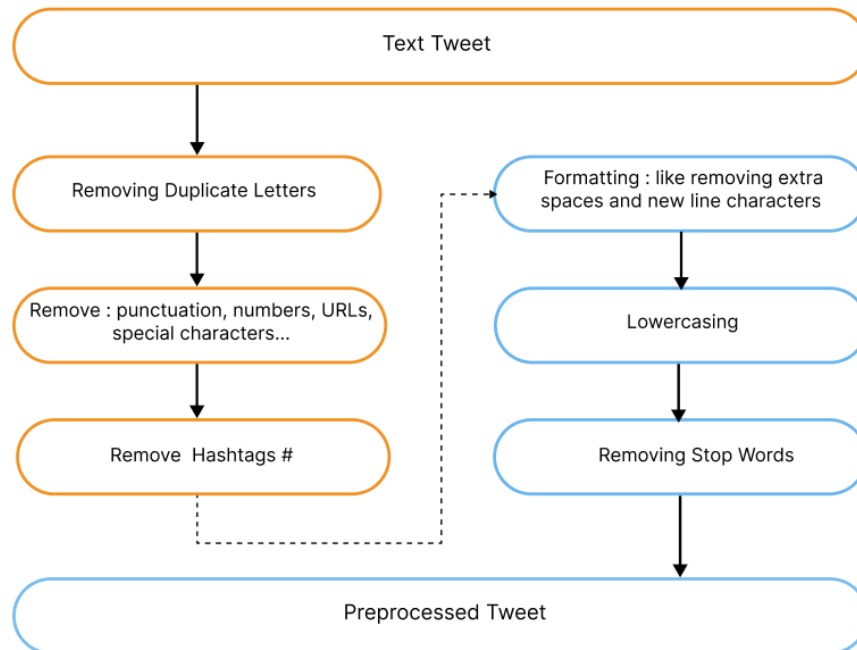


Figure 4.2: Basic steps of data preprocessing stage

2.3 BERT (Bidirectional Encoder Representations from Transformers)

In our work, we leverage tokenization and embedding techniques. Tokenization involves breaking down text into individual units (tokens), such as words or subwords, which BERT then processes. Embedding refers to the representation of these tokens in a continuous vector space, capturing their semantic meaning and relationships.

The key innovation of the BERT model [22] lies in its use of a bidirectional transformer encoder, enabling it to assimilate context from both directions simultaneously. As illustrated in Figure 4.3, BERT captures intricate word relationships within sentences by utilizing the efficient and parallelizable transformer architecture. Pre-trained on extensive text sources such as Wikipedia (2,500M words) and BookCorpus (800M words), BERT excels in understanding language nuances due to its training on diverse tasks like masked language modeling and next sentence prediction.

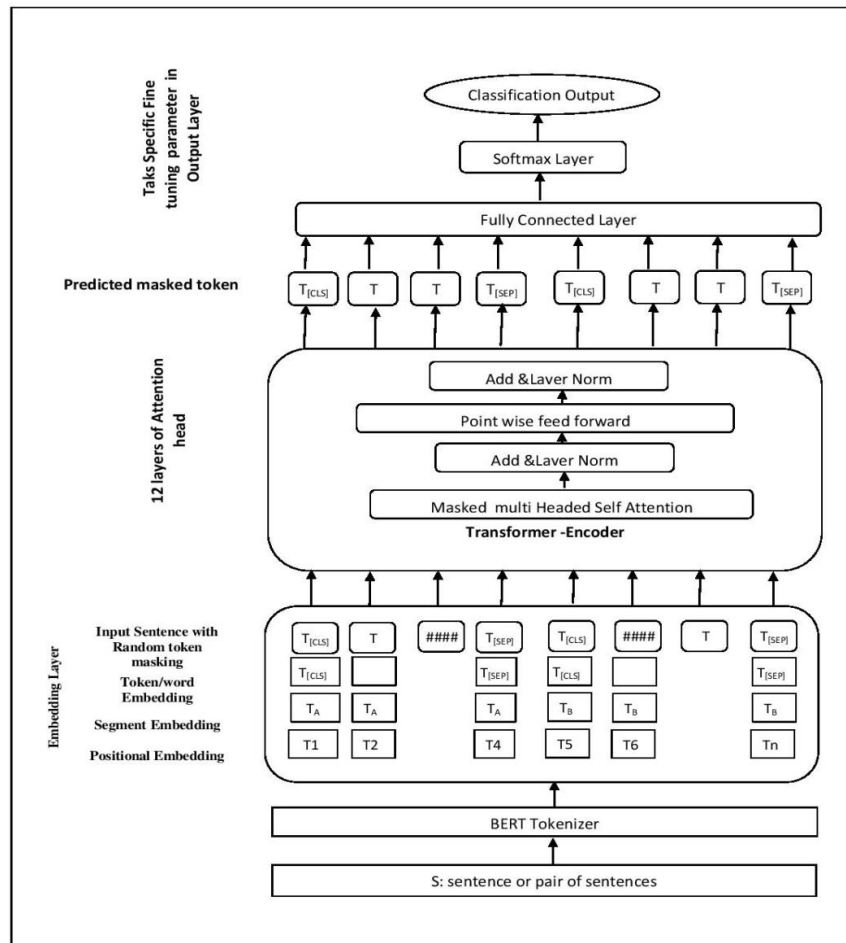


Figure 4.3: Bert Architecture [22]

2.3.1 Tokenization

Tokenization holds paramount importance in our work with BERT as it serves as the foundational preprocessing step in natural language processing tasks. By converting raw text data into a series of tokens, we enable BERT to effectively capture and understand linguistic nuances. The specialized tokenization strategy we employ ensures the efficient representation of textual information, facilitating robust language understanding and modeling with BERT.

```
tweet: 😞😞 مائيش مليحة
token_ids: [101, 100, 12441, 24148, 11626, 92313, 58754, 10382, 102]
```

Figure 4.4: Token ID

2.3.2 Generating Embeddings

In our process, generating embeddings involves converting input text into compact vector representations known as embeddings. These embeddings encode the contextual significance of words and their associations within a sentence. Token IDs play a pivotal role by serving as inputs into the BERT model. The primary output of BERT consists of contextualized embeddings for the input tokens, which encapsulate essential contextual details. These embeddings are versatile and can be used as inputs for various tasks.

```
[ 3.09326057e-03 -3.53989720e-01 5.13986945e-01 1.73380181e-01
 1.76747531e-01 9.02481228e-02 1.86970934e-01 1.52322561e-01
 3.05466838e-02 1.14312910e-01 2.23880962e-01 1.70052201e-01
 4.88828234e-02 1.37351245e-01 -1.52927963e-02 -2.42333010e-01
 1.52305618e-01 1.29314333e-01 -3.12141955e-01 3.73303920e-01
 -3.90709758e-01 1.76718943e-02 -1.28772855e-02 -2.53141463e-01
 2.98813760e-01 3.13672841e-01 -2.81088561e-01 3.00280564e-02
 1.35740399e-01 -3.09626251e-01 -4.63175550e-02 9.33116376e-02
 5.21714315e-02 3.73827159e-01 -9.92774591e-02 5.07881632e-03
 -6.58522487e-01 1.38938040e-01 1.14603281e-01 -2.42988437e-01
 -8.08537379e-02 -2.99963534e-01 -1.38957977e-01 3.59064974e-02
 -1.83967933e-01 2.43264914e-01 -1.08207138e-02 7.54893795e-02
 2.32168198e-01 4.43304598e-01 1.24663711e-01 ..... ]
```

Figure 4.5: Embeddings [17]

When utilizing BERT, it is generally unnecessary, and even discouraged, to conduct extensive preprocessing tasks like stop word removal or lemmatization. This is because BERT, being a pre-trained model, is specifically engineered to capture the contextual nuances of words and subwords within a sentence.

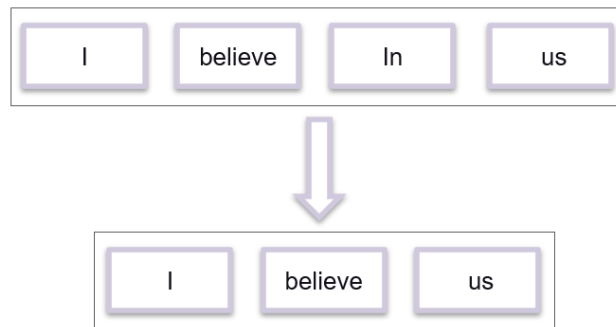


Figure 4.6: Extract Key Words and Omit Stop Words

2.4 Dimensionality Reduction

Many datasets contain a vast array of features, leading to complications such as heightened computational demands, model overfitting, and challenges in data visualization. To address these issues, we utilize Principal Component Analysis (PCA). This technique is widely employed technique for reducing the dimensionality of the data. Its objective is to identify the directions, termed principal components, along which the data exhibits the greatest variability. Subsequently, the data is projected onto a subset of these principal components, thereby alleviating the aforementioned challenges associated with high-dimensional datasets.

2.5 Ensemble Clustering

Ensemble clustering involves creating a set of diverse and accurate base clusterings or partitions from the given data. There are different methods to create an ensemble clustering. One common approach is to use Generation mechanisms. we are particularly interested in exploring this step by employing clustering on different object representation and different clustering algorithms and using the same algorithm with varied initializations or parameter settings.

2.5.1 Different Object Representation

In this mechanism, our focus revolves around handling both emojis and keywords to refine the emotional analysis of textual data. By addressing these two components, we aim to provide a more comprehensive understanding of the emotions conveyed within the text.

a. Emoji Handling : an emoji is a small digital image or icon typically used in electronic communication to express an idea, emotion, or concept. They are commonly used in messaging apps, social media platforms, and other forms of digital communication to add visual context and convey feelings or reactions in text-based conversations.

Eliminating an emoji from an emotion detection task is not the best course of action. Instead, we treat emojis in multiple ways to optimize the detection of the emotion present in the text.

Method 1: Replace emojis with their description using a python package (**emoji**):

```
"😂": "Face with Tears of Joy"
"😎": "Smiling Face with Sunglasses"
"😞": "Pensive Face"
"😭": "Crying Face"
"❤️": "Red Heart"
"💔": "Broken Heart"
"👐": "Palms Up Together"
```

Figure 4.7: Emojis description.

Method 2: We streamline the process by condensing a list of emojis conveying the same emotion into a single emoji through an emoji map sourced from PiliApp [76] and Emojipedia [30]. Then, we apply two different techniques to represent **the six basic emojis**.

```
emoji_mapping = {
"😄": ["😸", "😄", "🎂", "😊", "😂", "🤪", "👉", "😸", "😊", "👉"],
"😞": ["😞", "😞", "😞", "😞", "😞", "😞", "😞", "😞", "😞", "😞", "👤"],
"😱": ["😱", "😱", "😱", "😱", "😱", "😱", "😱", "😱", "😱", "😱", "👤"],
"😲": ["😲", "😲", "😲", "😲", "😲", "😲", "😲", "😲", "😲", "😲", "👤"],
"😡": ["😡", "😡", "😡", "😡", "😡", "😡", "😡", "😡", "😡", "😡", "👤"],
"😬": ["😬", "😬", "😬", "😬", "😬", "😬", "😬", "😬", "😬", "😬", "👤"],
}
```

Figure 4.8: Handling emojis with emoji map.

- **In the first technique**, we carefully substitute the resultant emoji with its corresponding meaning.

```
emojis = {
  " سعيد ": " 😊 ",
  " حزين ": " 😭 ",
  " متفاجئ ": " 😮 ",
  " خائف ": " 😱 ",
  " مقزز ": " 🤢 ",
  " غاضب ": " 😡 "}
```

Figure 4.9: Replace the emoji with its meaning(Arabic data).

```
emojis = {
  " happy " : " 😊 ",
  " sad " : " 😭 ",
  " surprise " : " 😮 ",
  " fear " : " 😱 ",
  " disgust " : " 🤢 ",
  " angry " : " 😡 "}
```

Figure 4.10: Replace the emoji with its meaning (English data).

- **In the second technique**, we transform the emoji into its corresponding textual description, utilizing the **demojize** function from the **emoji** library

```
" 😊 ": "grinning_face_with_smiling_eyes"
" 😭 ": "loudly_crying_face"
" 😮 ": "face_with_open_mouth"
" 😱 ": "face_screaming_in_fear"
" 🤢 ": "face_vomiting"
" 😡 ": "angry_face"
```

Figure 4.11: The six basic emojis with their descriptions.

Method 3

- Initially, we devised a function named **‘extract all emojis’**, This function extracts emojis also ensures that duplicates are removed while retaining their original order.
- Progressing to the classification phase, we established a new function called **‘classify emojis’**, tasked with categorizing the extracted emojis into predefined target emojis.
- Within the classification function, we initialized a dictionary to store the classified emojis. We then computed **the Euclidean distances** between the extracted emojis and the target emojis, relying on their **Unicode** points. For each extracted emoji, we determined the nearest target emoji and assigned it to the corresponding class in our **‘emoji classifications’** dictionary.

b. Extracting Affect Words :certain words convey emotions more strongly than others. However, let's take the sentence "Wafa received many new toys for her birthday" as an example. While the words "new," "toys," and "birthday" may not seem particularly emotional on their own, when combined, they evoke a sense of happiness.

We streamline the process by condensing a list of words conveying the same emotion into a single word through a keywords map sourced from **the NRC Emotion Lexicon**. Notably, the NRC Emotion Lexicon encompasses an extensive collection of 14154 keywords for each emotion. To manage this vast dataset effectively, we divide it into three partitions. With each iteration, we progressively integrate a new partition into the existing framework, meaning that the content of the first partition is included in partition 2, and the content of the second partition is included in partition 3.

```
emotion_keyword_mapping = {
  "angry" : ["idiotic", "offend", "strained", "punishment", "kicking", "hardened"],
  "disgust" : ["disobedience", "maggot", "betrayal", "ungrateful", "quagmire", "pungent"],
  "happy" : ["angelic", "jackpot", "pleasant", "amnesty", "aspire", "ardent"],
  "fear" : ["parachute", "horrified", "hopeless", "validity", "pare", "alertness"],
  "sad" : ["hydrocephalus", "infliction", "dull", "cross", "disqualify", "collapse"],
  "surprise": ["gawk", "improvisation", "excitation", "volcano", "cherish", "succeed"]
}
```

Figure 4.12: Sample Representative word set (English data).

```
emotion_keyword_mapping = {
  "غاضب" : ["متشدد", "ركل", "الثأر", "متوتر", "إهانة", "غبي"],
  "مقزز" : ["المستنقع", "جحود الجميل", "الخيانة", "البرقة", "العصيان"],
  "متحمس" : ["تطمح", "العفو", "ممتع", "الجائزة الكبرى", "ملائكي"],
  "خائف" : ["اليقظة", "القدر", "الصلاحية", "اليأس", "الرعب", "المظلة"],
  "حزين" : ["انهيار", "استبعاد", "صليب", "بليد", "إصابة", "استسقاء الرأس"],
  "متفاجئ" : ["نجاح", "نعتز به", "بركان", "إثارة", "ارتجال", "تحديق"]
}
```

Figure 4.13: Sample Representative word set (Arabic data).

Note : it is important to note that in our analysis of Arabic data, we employed the mechanisms using both English and Arabic keywords. This approach is particularly crucial because the Arabic data, specifically in the Algerian dialect, encompasses multiple languages. By doing so, we can more effectively manage the linguistic diversity and accurately capture the emotional nuances present in the multilingual Algerian context.

2.5.2 Different clustering algorithms

In this mechanism, we focus on combining multiple clustering algorithms, specifically k-means, Gaussian mixture models, and agglomerative clustering to enhance the emotion detection of textual data. By leveraging the strengths of different algorithms, we aim to improve the accuracy and robustness of emotion detection. This mechanism allows us to better capture the nuances and complexities of emotions in text, leading to a more comprehensive understanding of the underlying emotional content.

Let's examine the reasoning behind choosing these algorithms. We'll highlight the specific factors that influenced our selection:

a. K-means Clustering:

- **Efficient and Simple:** Ideal for large datasets, helping categorize emotions clearly.
- **Centroid-Based:** Captures spherical clusters of similar emotions.

b. Gaussian Mixture Models (GMM):

- **Flexible:** Models clusters of various shapes, capturing subtle emotional variations.
- **Soft Clustering:** Reflects overlapping emotions by allowing data points in multiple clusters.

c. Agglomerative Clustering:

- **Hierarchical:** Provides detailed views of emotional groupings.
- **Adaptive:** No need for a predefined number of clusters, suitable for diverse emotions.

2.5.3 Different parameter initialization

In this mechanism, we focus on exploring different parameter initialization methods for clustering algorithms to enhance the emotion detection of textual data. We have applied this mechanism specifically to k-means and Gaussian mixture models (GMM) to examine the effects of various initialization techniques on these algorithms.

2.6 Evaluation Metrics

Clustering serves as a method to identify similarities among data points lacking predefined class labels. It partitions the data into multiple clusters, ensuring that data points within the same cluster exhibit higher similarities compared to those in different clusters.

As clustering operates within unsupervised learning, it lacks inherent means to validate model accuracy. To address this limitation, various methods have been devised, including: internal evaluation, external evaluation and manual evaluation.

In our study, we employ **internal metrics** to assess the performance of clustering algorithms. These metrics allow us to evaluate the quality of clusters formed by algorithms without relying on external class labels. Internal metrics provide valuable insights into the effectiveness of clustering techniques by analyzing the cohesion and separation of data points within clusters. Some common types of internal metrics that we consider include:

2.6.1 Silhouette Coefficient

The silhouette coefficient is calculated for each data point using mean intra-cluster distance and mean inter-cluster distance. [100]

$$SilhouetteCoefficient = \frac{b - a}{\max(a, b)}$$

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, with -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.

The closer the value is to 1, the better the clustering method.

2.6.2 Davies-Bouldin Index

Davies-Bouldin Index [100] can be calculated as follows:

$$Davies - BouldinIndex = \frac{1}{c} \sum_{i=1}^c \max_{j \neq i} \frac{\sigma_i + \sigma_j}{d(c_i, c_j)}$$

where c is the number of clusters, σ_i is the dispersion of cluster i , σ_j is the dispersion of cluster j , and $d(c_i, c_j)$ is the distance between the centroids of clusters i and j .

In practice, lower DBI values indicate better clustering performance, whereas higher values indicate poorer performance.

2.6.3 Calinski-Harabasz Index (CHI)

Calinski-Harabasz Index [100] (also called variance ratio criterion) is the ratio between between-cluster dispersion and within-cluster dispersion for all clusters.

$$\text{Calinski-Harabasz Index} = \frac{\text{trace}(B_c)}{\text{trace}(W_c)} * \frac{n_E - c}{c - 1}$$

where, \mathbf{c} is the number of cluster, n_E is the size of dataset E , and $\text{trace}(\mathbf{Bc})$ is the trace of between-cluster (inter-cluster) dispersion matrix, and $\text{trace}(\mathbf{Wc})$ is the trace of within-cluster (intra-cluster).

higher values indicate better-defined clusters and thus better clustering performance, while lower values indicate the opposite.

3 Implementation

In this section, we provide a concise overview of the programming environment and language employed in building our system. Furthermore, we delve into the implementation intricacies of various components within our system, offering a comprehensive understanding of its development process.

3.1 Programing Environment

To develop our system, we utilized a variety of tools. In the following section, we outline and define the materials and resources employed at each stage of our process. This includes the specific software, libraries, and methodologies that facilitated the execution and optimization of our work. By detailing these tools, we aim to provide a comprehensive overview of the technical foundation supporting our system's development and implementation.

3.1.1 Google Colaboratory

we utilized Google Colaboratory [39] (Google Colab), a cloud-based platform that offers a robust environment for Python coding and data analysis. Colab's key features include cloud accessibility, seamless integration with Google Drive, pre-installed popular libraries, and free access to powerful GPUs and TPUs. Its collaborative capabilities and interactive coding environment facilitated efficient development and experimentation. These features made Google Colab an indispensable tool for optimizing and implementing our project.

3.1.2 Python language

Python [78] was essential in developing our system due to its versatility, simplicity, and robust library ecosystem. Its readability and ease of use accelerated our development process, while libraries such as NumPy, pandas and scikit-learn provided powerful tools for data manipulation, machine learning, and deep learning. Python's widespread use in the scientific and data analysis communities ensured strong support and extensive resources, making it the ideal choice for our project.

3.1.3 Scikit-learn

Scikit-learn [73] is a powerful Python library widely used for machine learning and data analysis. It provides simple and efficient tools for data mining and data analysis, built on top of NumPy, SciPy, and matplotlib. Scikit-learn offers a range of supervised and unsupervised learning algorithms, including clustering, regression, and classification. Its easy-to-use interface and well-documented code make it an ideal choice for both beginners and experienced practitioners in developing robust machine learning models and performing complex data analysis tasks.

3.1.4 NumPy

NumPy [40], short for Numerical Python, is a fundamental library for scientific computing in Python. It provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy is essential for performing numerical calculations, linear algebra, Fourier transforms, and random number generation. It serves as the backbone for many other scientific computing

libraries in Python, such as SciPy, pandas, and scikit-learn, by enabling high-performance operations on large datasets.

3.1.5 Pandas

Pandas [60] is a powerful and versatile data manipulation library in Python, designed for handling and analyzing structured data. It provides data structures like Series (one-dimensional) and DataFrame (two-dimensional) that make it easy to manage, clean, and analyze data. Pandas excels in data manipulation tasks, including data cleaning, transformation, and aggregation, making it a favorite tool for data scientists and analysts. With its intuitive syntax and robust functionality, Pandas simplifies tasks like merging datasets, handling missing data, and performing complex data operations, thereby streamlining the data analysis workflow.

3.1.6 Matplotlib

Matplotlib [59] is a versatile Python library for creating high-quality visualizations, including line plots, scatter plots, histograms, and more. It offers extensive customization options and integrates seamlessly with other Python libraries like NumPy and Pandas. With Matplotlib, users can generate static, animated, and interactive plots to effectively visualize data in scientific computing, data analysis, and machine learning projects.

3.1.7 NLTK

NLTK [13], or Natural Language Toolkit, is a powerful Python library designed for working with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources, along with a suite of text processing libraries for tasks such as tokenization, stemming, tagging, parsing, and classification. NLTK is widely used in various natural language processing (NLP) applications, including sentiment analysis, machine translation, and information retrieval.

3.2 Tweets Tokenization and Embedding

Now, we will explain our method for converting text data into numerical form. For emotion detection, we selected the BERT Multilingual Base Model from the popular Hugging Face platform, which is a type of Transformer model.

3.2.1 Hugging face

Hugging Face [31] is an artificial intelligence company and an open-source community that specializes in natural language processing (NLP) technologies. The company provides a comprehensive platform and library for developing and deploying machine learning models, particularly those based on transformers, which are advanced architectures for NLP tasks. Hugging Face is widely known for its Transformers library, which includes pre-trained models for a variety of languages and tasks such as text classification, sentiment analysis, translation, and question answering. The platform aims to make NLP more accessible and easier to integrate into applications by offering tools, datasets, and model hubs that support cutting-edge research and development.

3.2.2 Sentence Transformer

This framework [82] offers a simple way to generate dense vector representations for sentences, paragraphs, and images. The models, built on transformer networks such as BERT, RoBERTa, and XLM-RoBERTa, deliver state-of-the-art performance across various tasks. Text is embedded into vector space, ensuring that similar texts are positioned closely together, facilitating efficient retrieval using cosine similarity.

3.2.3 Transformers

Transformers [83] offers APIs that allow you to quickly download and apply pretrained models to your text data, fine-tune them on your own datasets, and share them with the community via our model hub. Additionally, each Python module defining a model architecture is completely standalone and can be easily modified for rapid research experiments.

3.2.4 BERT Multilingual Base Model

BERT [24] is a transformers model pretrained with two objectives: masked language modeling (MLM) and next sentence prediction (NSP). MLM enables the model to learn bidirectional representations of sentences, while NSP predicts whether two sentences follow each other. These sentence pairs can be either consecutive sentences from the original text or unrelated ones, enhancing the model's understanding of sentence relationships. The figure below present the used python script to import and initialize the Bert based tokenazation model.

imports and initializes the essential components for using the BERT-based model and tokenizer to generate tweet embeddings

```
from transformers import AutoTokenizer, AutoModel
from sentence_transformers import SentenceTransformer

tokenizer = AutoTokenizer.from_pretrained("bert-base-multilingual-cased")
model = AutoModel.from_pretrained("bert-base-multilingual-cased")
embedder = SentenceTransformer('bert-base-multilingual-cased')
```

Figure 4.14: Imports and initializes BERT-based model and tokenizer.

When utilizing the `AutoTokenizer` class, tokenization is further optimized through the utilization of a pretrained model specifically engineered to handle multilingual text while preserving casing information. This ensures that BERT can effectively process text from diverse linguistic contexts.

```
cleaned_tweets_keywords = [replace_keywords_with_emotions(tweet) for tweet in processed_tweets]

# tokenize data
Tokenized_tweets = [tokenizer.encode(tweet, add_special_tokens=True) for tweet in cleaned_tweets_keywords]

# data embedding
Tweet_embeddings = embedder.encode(Tokenized_tweets, show_progress_bar=True)
print(Tweet_embeddings.shape)
```


Batches: 100%  314/314 [00:07<00:00, 63.71it/s]
(10017, 768)

Figure 4.15: Tokens Embedding.

The `encode()` method plays a pivotal role in the BERT encoding process by transforming raw text inputs into symbolic sequences represented as integers. This method not only generates token sequences from the model's vocabulary but also incorporates special tokens such as `[CLS]` and `[SEP]`. The `[CLS]` token, positioned at the beginning of the

sequence, indicates the start of a classification task, guiding the model to process input for classification purposes. Conversely, the '[SEP]' token is utilized to delineate different segments within the input text, marking the end of a segment or sentence. Additionally, the 'encode()' method manages padding, ensuring uniformity in sequence length across inputs within a batch.

3.3 Dimensionality Reduction

The output shape of the Bert model was (4134, 768) for Arabic data and (10017, 768) for English data. This size, being relatively large, poses challenges during the clustering step. To address this, we utilized PCA to effectively reduce dimensionality while preserving as much variance as possible. This approach enables us to maintain a compact representation of the data without sacrificing crucial patterns or structures.

```
▶ from sklearn.decomposition import PCA

# Apply PCA for dimensionality reduction
pca = PCA(n_components=2) # Specify the desired number of components
reduced_features = pca.fit_transform(Tweet_embeddings)

print(reduced_features)
```

```
⇒ [[-2.278775    0.12814598]
   [-2.2787707  0.12808035]
   [-2.2787707  0.12807593]
   ...
   [ 2.5964315 -0.7199363 ]
   [ 2.5964315 -0.7199363 ]
   [ 2.5964315 -0.7199364 ]]
```

Figure 4.16: PCA Tweets Embedding Dimensionality Reduction.

3.4 Ensemble Clustering

We utilized our original data after reducing their dimensions in the previous step. The Figure 4.17 represents our ensemble clustering class, where we've defined three main functions ('__init__()', 'fit()', and 'predict()'). In '__init__()', we specify parameters such as

the number of data splits, the cluster number, and the estimators that will contain the models fit. The `fit()` function integrates the clustering algorithm models. Notably, the number of splits also serves as a parameter in the `for` loops, indicating the frequency of algorithm runs. In `predict()`, a majority voting technique is employed to choose the final prediction from the ensemble of models.

```
class EnsembleClustering:
    def __init__(self, n_clusters=6, n_splits=10):
        self.n_clusters = n_clusters
        self.n_splits = n_splits
        self.estimateds = []

    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):
            kmeans = KMeans(n_clusters=self.n_clusters, init='k-means++', n_init=10, max_iter=300, random_state=i)
            X_subset = embeddings_splits[i]
            kmeans.fit(X_subset)
            self.estimateds.append(kmeans)

    def predict(self, X):
        cluster_assignments = np.zeros((X.shape[0], self.n_splits))
        for i, estimator in enumerate(self.estimateds):
            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 4.17: Ensemble clustering class.

We instantiate an object from the ensemble clustering class, employing the specified parameters: six clusters and 22 splits. The subsequent demonstration, as depicted in Figure 4.18, showcases the application of our ensemble clustering approach to the provided data, including the labeling prediction process.

```
X, y = make_blobs(n_samples=93, centers=6, random_state=42)
embeddings = reduced_features

ensemble_clustering = EnsembleClustering(n_clusters=6, n_splits=22)
ensemble_clustering.fit(embeddings)

y_pred2 = ensemble_clustering.predict(embeddings)

# Mapping cluster labels to emotions
emotion_mapping = {0: 'angry', 1: 'happy', 2: 'sad', 3: 'fear', 4: 'surprise', 5: 'disgust'}
ensemble_label1=[]

# Convert cluster labels to emotion labels
ensemble_labels = [emotion_mapping[label] for label in y_pred2]

# Print predicted labels for each tweet
for i, label in enumerate(y_pred2):
    emotion_label = emotion_mapping[label]
    ensemble_label1.append(emotion_label)
    #print(f"Tweet {i + 1}: {emotion_label}")
print(ensemble_label1)

# Plot the scatter plot with truncated cluster labels
plt.scatter(X[:, 0], X[:, 1], c=y_pred2[:len(X)], cmap='viridis')
plt.colorbar(label='Cluster Label')
plt.title('ensemble clustering with description emojis')
plt.show()
```

Figure 4.18: Create instance of ensemble clustering class and fit the data.

3.5 Analysis of English Data

In this section, we will present the English data analyze using ensemble clustering techniques . Starting with an overview of the dataset and its features, followed by the preprocessing steps taken to clean and prepare the data. Next, we will present the results of our analysis including a discussion interpreting these findings.

3.5.1 Dataset

The features of textual data, essential for emotion detection, profoundly influence the accuracy and efficacy of our analysis, particularly considering the differences between English and Arabic datasets. In the subsequent sections, we introduce the specific features of the English data.

- The Figure 4.19 illustrating key features of the English data, showcasing word count, emoji distribution, and specific emoji occurrences.

```
Total words: 131265
Total emojis: 4806
Percentage of emojis compared to words: 3.66%
Occurrences of specific emojis:
😄 : 734
😡 : 494
😱 : 32
😞 : 14
😏 : 194
😬 : 26
Total other emojis: 3312
Percentage of specific emojis compared to other emojis: 45.11%

Number of tweets with emojis: 2257
Number of tweets without emojis: 7760
```

Figure 4.19: Exploring English Data Features: Insights into Emojis.

Note : on the one hand ,our analysis focuses on handling emojis, despite the data showing that the number of emojis is significantly less than the number of words . However, it's important to recognize that the impact of emojis is substantial. A single emoji can effectively express an emotion that would otherwise require a set of words. This highlights the importance of considering emojis in our emotional analysis, as their ability to convey complex emotions concisely enhances our understanding of emotional expressions within the dataset.

In addition to emoji distribution, We extracted emotion words counts directly from the original English data:

- Emotion words from original English data: {'angry': 157, 'disgust': 22, 'happy': 454, 'fear': 82, 'sad': 235, 'surprise': 56}

The analysis also reveals the prevalence of specific emotions within the English data by keywords identified and their respective frequencies documented:

- Keywords for angry = 5058
- Keywords for happy = 4147

- Keywords for sad = 4707
- Keywords for surprise = 22097
- Keywords for disgust = 6425
- Keywords for fear = 4521

3.5.2 Preprocessing

To implement the preprocessing steps outlined in our conception, we utilized several packages. We will now discuss some of them, along with detailing some of the functions we used from each, as shown in Figure 4.20.

- **Emoji:** The “**emoji**” package [52] is a Python library that allows for easy handling and manipulation of emojis in text, enabling the addition, removal, and conversion of emojis within strings.
- **RegEx:** Regular Expression [10] is a powerful tool used for pattern matching and manipulation within strings. It allows for searching, replacing, and extracting specific patterns of text, making it essential for tasks involving text processing and data validation.

```

def Preprocess(tweet):
    # Remove URLs
    tweet = re.sub(r'http\S+', '', tweet)
    # Remove mentions
    tweet = re.sub(r'@\w+', '', tweet)
    # Remove hashtags
    tweet = re.sub(r'#\w+', '', tweet)
    # remove new line \n
    tweet = re.sub(r'\n', '', tweet)
    # removing the '_'
    tweet = re.sub(r'_', ' ', tweet)
    # Remove special characters and punctuation
    tweet = re.sub(r'^a-zA-Z\s\u0001F600-\u0001F64F\u0001F300-\u0001F5FF\u0001F680-\u0001F6FF\u0001F700]+', '', tweet)
    # Remove numbers
    tweet = re.sub(r'\d+', '', tweet)
    # Convert to lowercase
    tweet = tweet.lower()
    # Remove extra spaces
    tweet = re.sub(r'\s+', ' ', tweet).strip()
    # # Remove stop words
    # stop_words = set(stopwords.words('english'))
    # tweet = ' '.join([word for word in tweet.split() if word not in stop_words])
    return tweet

```

Figure 4.20: Script of Pre-processing Functions.

```

from transformers import BertTokenizer, BertModel
import pandas as pd
import emoji
from sklearn.decomposition import PCA
# Load tweet data and tokenize the text
data = pd.read_csv('/content/drive/MyDrive/Datasets/English_data_unlabeled.csv')

processed_tweets = [Preprocess(tweet) for tweet in data["Tweets"]]
cleaned_tweets_description = [emoji.demojize(tweet) for tweet in processed_tweets]

print(cleaned_tweets_description[2])

```

ed happy birthday to one smokin hot mama :hot_face: i love you so much lil youre an amazing friend

Figure 4.21: Read and Clean the Data.

3.5.3 Results and discussion

In this section, we present the results of various ensemble clustering mechanisms. Each approach involves creating diverse base clusterings through different object representations, clustering algorithms, and varied initializations or parameter settings. The following subsections detail the outcomes of these mechanisms, highlighting their effectiveness.

1. Results of different object representations : In Table 4.1 , we explore the results of using different object representations with KMeans clustering ensembles. The parameters used for these ensembles are:

Kmeans clustering ensemble : (n clusters=6, n splits=22) [KMeans(n clusters=6,in t='k-means++', n init=10, random state=42)]

Note:

- **K1** : 4718 words,
- **K2** : 9436 words,
- **K3** : 14154 words.
- **S** : Silhouette Score,
- **C** : Calinski-Harabasz Score,
- **D** :Davies-Bouldin Score

Table 4.1: Result of KMeans clustering ensembles in English data

	Techniques	Silhouette Score	Calinski-Harabasz Score	Davies-Bouldin Score
Keywords	K1	0.63551307	14652.3331413960	3.1925711155368304
	K2	0.8201926	91745.62641078627	0.3754570366450628
	K3	0.6870811	29289.441086531213	0.7335313983006653
Emojis description (D)	D	0.50868887	48407.27635915183	0.7517529256552749
	D+K1	0.7347591	130015.38021253291	0.5638244511469562
	D+K2	0.10922659	869.2569332902694	1.4192646688518427
	D+K3	0.73475957	130015.62212919841	0.5638228414703661
Emojis map & meaning (M)	M	0.68193233	48837.38576467099	0.6436907611831766
	M+K1	0.7980847	74165.40388417376	0.4885577464721909
	M+K2	0.06491469	817.1375810723973	2.4746329587585048
	M+K3	0.79808414	74165.220743793	0.4885585394739304
Emojis map & description (MD)	MD	0.5086886	484007.22746450475	0.751752863345457
	MD+K1	0.734759	130015.56119457242	0.5638224208493109
	MD+K2	0.10922663	869.256640416271	1.419264762233445
	MD+K3	0.7347591	130015.32629849388	0.5638230763300406

The Silhouette Scores for Keywords (K1, K2, K3) alone are relatively good, indicating decent clustering quality. However, combining Keywords with other techniques often results in even better clustering performance. For example, the Silhouette Score for K1 alone is 0.63551307, but it increases to 0.7347591 with D+K1, 0.7980847 with M+K1, and 0.734759 with MD+K1. Among these combinations, the best results are achieved with Emojis map & meaning (M) combined with Keywords, as seen with the highest Silhouette Score of 0.7980847 for M+K1. This suggests that integrating Keywords with other methodologies, particularly emoji map & meaning, significantly enhances clustering performance with k-means ensemble clustering.

In our pursuit of achieving optimal results across various object representations, we applied ensemble clustering using two distinct algorithms: Agglomerative Clustering and Gaussian Mixture Models (GMM). These algorithms were specifically applied to the best-performing results obtained from **Table 4.1**.

For Agglomerative clustering ensemble : (n clusters=6, n splits=22) [Agglomerative (n components=6, linkage = 'ward')].

For Gaussian Mixture Models clustering ensemble : (n clusters=6, n splits=22) GMM (n components=6, covariance type='diag').

Table 4.2: Ensemble clustering using Agglomerative and GMM in English

Technique	Metrics	GMM	Agglomerative
K2	S	0.21689884	0.80711025
	C	1293.23298 1595158	82095.06365 364435
	D	4.17938973 8257317	0.43821283 64154188
D+K3	S	0.18308505	0.74913967
	C	1090.5840789 787251	155886.54930 238615
	D	1.293776685 1095973	0.48956845 0981449
M+ K1	S	0.5722231	0.7871192
	C	13944.01990 8610948	62672.508411 550356
	D	1.122246204 9973974	0.442249029 9139839
MD+K1	S	0.3553664	0.79646826
	C	3971.337859 7081446	77762.77730 348012
	D	1.54590088 02890442	0.40841224 88633794

The Table 4.2 demonstrates that Agglomerative clustering outperforms GMM across all techniques. The higher Silhouette Scores and Calinski-Harabasz Scores, alongside the lower Davies-Bouldin Scores, confirm that Agglomerative clustering forms more coherent and distinct clusters compared to GMM. The exceptional Silhouette Score of 0.80711025 for the K2 technique with Agglomerative clustering underscores the algorithm's robustness and effectiveness in ensemble clustering tasks.

In Figure 4.22, we present the results of Method 3, which yielded suboptimal solutions. Consequently, we opted not to incorporate it into our ensemble clustering approach.

Classified as 😊 : [' 😊 ', ' 😐 ', ' 😊 ', ' ❤️ ', ' 📅 ']
 Classified as 😭 : [' 😭 ', ' 😬 ']
 Classified as 🤔 : [' 🤔 ', ' 😬 ', ' 🤔 ']
 Classified as 😲 : [' 😲 ', ' 😲 ']
 Classified as 😡 : [' 😡 ', ' 😬 ', ' 😭 ']

Figure 4.22: Result of method 3.

2. Results of different clustering algorithms : In Table 4.3, we present the results of using different clustering algorithms with specified parameters to create ensemble clusterings. These algorithms were specifically applied to the best-performing results obtained from the different object representation mechanisms. The parameters for these ensembles are **n clusters=6** and **n splits=22**. We employed the following combinations:

- a. **Ensemble k-means, mixture, and agglomerative clustering:** - GMM (n components=6, covariance type='diag') - KMeans (n clusters=6, init='k-means++', random state=42) - Agglomerative (n components=6, linkage='ward')
- b. **Ensemble k-means and agglomerative clustering:** - KMeans (n clusters=6, init='k-means++', random state=42) - Agglomerative (n components=6, linkage='ward')
- c. **Ensemble k-means and mixture clustering:** - GMM (n components=6, covariance type='diag') - KMeans (n clusters=6, init='k-means++', random state=42)
- d. **Ensemble mixture and agglomerative clustering:** - GMM (n components=6, covariance type='diag') - Agglomerative (n components=6, linkage='ward')

Table 4.3: Result of Ensemble clustering in English data

Technique	Metrics	Ensemble of 3 Algorithmes	K-means+ Agglomerative	KMeans+ Gaussian	Gaussian+ Agglomerative
K2	S	0.8082085	0.80820864	0.011694864	0.80820876
	C	79735.94 280219184	79736.031 53231436	920.8410 284863651	79736.5010 7745678
	D	0.3684250 721751218	0.3684243 4308936906	2.9568584 45295924	0.36842543 818738943
D+K3	S	0.7274913	0.7815624	0.6019943	0.72749186
	C	130331.28 169056919	84398.430 14381525	62697.651 13988638	130331.459 96157857
	D	0.5378197 632860099	0.4947964 9837330358	0.4661196 4089581764	0.537816976 7942383
M+K1	S	0.5369295	0.7792044	0.71646714	0.7792044
	C	14061.808 378101427	63916.362 947283305	32581.244 80022501	63915.18997 27321
	D	1.31400280 0527694	0.4569729 055382892	1.2430605 487306219	0.456977362 4310516
MD+K1	S	0.77386034	0.78156227	0.77142	0.78156227
	C	51658.502 12584226	84398.3933 3180367	55432.819 606339435	84398.32007 944134
	D	0.5199250 944519281	0.49479724 51307691	0.52894480 89657653	0.494797308 73461497

It is evident that the combination of K-means+Agglomerative consistently performs well across various metrics. However, the highest specific result achieved is 0.80820876, which comes from the Gaussian+Agglomerative technique. This result is only marginally better than the best result obtained from the K-means +Agglomerative combination, indicating that while the Gaussian+Agglomerative combination slightly outperforms in this instance, the K-means+Agglomerative method remains a robust and reliable choice overall.

3. Results of different parameter initialization : In Table 4.4, we delve into the results obtained by employing various parameter initializations in ensemble clustering. These approaches were specifically applied to the method yielding the best results among different clustering algorithms. Additionally, we introduced other methods to further enhance our analysis. The parameters for these ensemble methods remain consistent, with **n clusters=6**

and $n \text{ splits}=22$. The following combinations were employed:

a. Ensemble 2-KMeans and GMM

KMeans (n clusters=6, init='k-means++', random state=42),

GMM (n components=6, covariance type='diag'),

KMeans (n clusters=6, init='center', random state=42).

b. Ensemble 2-Agglomeratives and GMM

Agglomerative (n components=6, linkage='ward'),

GMM (n components=6, covariance type='diag'),

Agglomerative (n components=6, linkage='complete').

Table 4.4: Results of different parameter initialization in English

Technique		2 k-means + GMM	2 Agglomerative + GMM
K2	S	0.8202998	0.5521399
	C	93012.11590289751	8362.180473101296
	D	0.3782209000635028	0.8169632690347536
D+K3	S	0.773124	0.08492206
	C	108928.95911914934	3677.653047998628
	D	0.5156521856574553	2.4402174540779726
M+K1	S	0.78001547	0.7236625
	C	67938.16761191162	35752.00288181489
	D	0.5056122373862173	0.43095700666584946
MD+K1	S	0.7917687	0.29623052
	C	89848.14489537146	3863.037036765572
	D	0.4977722971592023	1.2277845738242374

It is clear that the "2 K-means + GMM" combination consistently outperforms the "2 Agglomerative + GMM" combination. Specifically, the highest score achieved is 0.8202998, obtained from the "2 K-means + GMM" method. This indicates a significant improvement in clustering performance compared to the best result from the "2 Agglomerative + GMM" combination, which is 0.5521399. The "2 K-means + GMM" technique not only yields the highest score but also shows robust performance across various metrics, making it a superior choice for clustering in this context.

3.6 Analysis of Arabic Data

In this section, we analyze the Arabic data using ensemble clustering techniques . We start with an overview of the dataset and its features, followed by the preprocessing steps taken to clean and prepare the data. Next, we present the results of our analysis and conclude with a discussion interpreting these findings.

3.6.1 Dataset

We will explore the distinct characteristics of the Arabic dataset, highlighting its unique features and their contributions to our analysis. It's important to note that Arabic (Algerian dialect) presents notable differences compared to English, which can significantly influence the outcomes of our study.

- ☞ The Figure 4.23 illustrating key features of the Arabic data, showcasing word count, emoji distribution, and specific emoji occurrences.

```
Total words: 24964
Total emojis: 3023
Percentage of emojis compared to words: 12.11%
Occurrences of specific emojis:
😄 : 775
😂 : 185
😱 : 21
😬 : 34
😡 : 77
😭 : 8
Total other emojis: 1923
Percentage of specific emojis compared to other emojis: 57.20%

Number of tweets with emojis: 919
Number of tweets without emojis: 3215
```

Figure 4.23: Exploring Arabic Data Features: Insights into Emojis.

- ☞ In addition to emoji distribution, We extracted emotion words counts directly from the original Arabic data:
 - Emotion words from original arabic data: {'surprised': 0, 'sad': 0, 'fear': 0, 'happy': 25, 'disgust': 0, 'angry': 0}

☞ The analysis also reveals the prevalence of specific emotions within the Arabic data by English and Arabic keywords identified and their respective frequencies documented:

Arabic keywords:

- Keywords for angry = 24
- Keywords for happy = 14
- Keywords for sad = 9
- Keywords for surprise = 2297
- Keywords for disgust = 10
- Keywords for fear = 26

English keywords:

- Keywords for angry = 0
- Keywords for happy = 25
- Keywords for sad = 7
- Keywords for surprise = 1
- Keywords for disgust = 0
- Keywords for fear = 0

3.6.2 Preprocessing

To implement the preprocessing steps for the Arabic data, we utilized the same packages as for the English data, with an additional package specifically for handling Arabic text. We will now discuss this package, and detail the functions we used from it.

- **PyArabic** :[\[2\]](#) is a Python library tailored for managing Arabic text, offering functionalities like text normalization, stemming, and tokenization. Notably, it includes functions such as `'strip_tashkeel'` and `'strip_tatweel'`, which facilitate tasks like removing diacritics and elongation marks, enhancing the processing of Arabic language data.

```
# Remove diacritics
tweet = strip_tashkeel(tweet)
# Remove tatwil alharf
tweet = strip_tatweel(tweet)
# remove duplicate letters
tweet = remove_duplicate_letters(tweet)
```

Figure 4.24: Pre-processing additional Functions.

3.6.3 Results and discussion

In this section, we unveil the outcomes of several ensemble clustering mechanisms for Arabic data. Each approach encompasses the generation of diverse base clusterings employing various object representations, clustering algorithms, and diverse initializations or parameter configurations. Subsequent subsections meticulously elucidate the results of these methodologies.

1. Results of different object representations : In Table 4.5 & 4.6, we explore the outcomes of utilizing various object representations with KMeans clustering ensembles for Arabic data. Notably, the parameters utilized for these ensembles align with those of the English data. It's important to note that separate results are presented for Arabic and English keywords.

Table 4.5: Result of KMeans clustering ensembles in Arabic data with Arabic keyword

	Techniques	Silhouette Score	Calinski-Harabasz Score	Davies-Bouldin Score
Keywords	K1	0.44243714	1654.5005880625558	0.753867147224315
	K2	0.095876314	1684.2500919278755	2.05560327295802
	K3	0.2314722	3222.84909497148	1.18388424901961
Emojis description (D)	D	0.4847758	3367.769010987639	2.1279849724660944
	D+K1	0.3865655	13656.377393953721	0.8958112650861715
	D+K2	0.2603351	2293.6786762732513	0.964487660690156
	D+K3	0.123009235	1793.5544287651603	0.9273953941726886
Emojis map & meaning (M)	M	0.713531	20809.897109448524	0.45602117108059553
	M+K1	0.37672126	5194.5751409701692	0.6885779524319074
	M+K2	0.34749	2605.13578870511	1.4515131858678014
	M+K3	0.54133844	14267.479035497514	0.731891815737962
Emojis map & description (MD)	MD	0.4847758	3367.7684560691996	2.127984828853112
	MD+K1	0.38656536	3656.3803138359863	0.8958114176920924
	MD+K2	0.26033512	2293.6785522373625	0.9644876081180385
	MD+K3	0.12300883	1793.5589861751657	0.9273937200898096

The table 4.5 demonstrates that the most effective approach for clustering in this context appears to be using emojis map and meaning without additional keyword data. This technique achieves the highest Silhouette Score of 0.713531, indicating well-defined and effective clustering. The lower performance of keywords and their combinations may be due to the multilingual nature of the Algerian dialect in the data, which includes elements of Arabic, French, and other languages, while the keywords are in pure Arabic. This linguistic mismatch likely reduces the effectiveness of the keyword-based techniques.

Table 4.6: Result of KMeans clustering ensembles in Arabic data with English keyword

	Techniques	Silhouette Score	Calinski-Harabasz Score	Davies-Bouldin Score
Keywords	K1	0.67778635	13930.802059814221	0.4904654077827864
	K2	0.7015065	15019.922494875575	0.7243414556260191
	K3	0.31340364	1414.038289123046	0.778828323010245
Emoji description (D)	D	0.48477587	3367.7684973138307	2.129785233120797
	D+K1	0.44216302	3695.3793659356315	1.5349932538489732
	D+K2	0.3857908	5364.913960916501	0.9883121078747635
	D+K3	0.10213358	1798.5353851181935	1.1863928001241444
Emojis map & meaning (M)	M	0.71353114	20809.91651032707	0.4560211346670284
	M+K1	0.72585523	22248.16336927542	0.4537078290501955
	M+K2	0.69142765	17753.009147161778	0.5780434912623534
	M+K3	0.399814	1895.8728659476915	1.0548985784928782
Emojis map & description (MD)	MD	0.484776	3367.7705969073213	2.1279848488423267
	MD+K1	0.44216287	3695.3759499829775	1.5349938940384573
	MD+K2	0.38579088	5364.9173176114555	0.9883117484306002
	MD+K3	0.102133326	1798.534147224279	1.18639383433189218

The **Table 4.6** demonstrates that the most effective approach for clustering in this context is using the combination of emojis map & meaning with the K1 English keyword set (M+K1), which achieved the highest silhouette score of 0.72585523. The high performance of K1 when combined with emojis map & meaning, suggests that integrating well-chosen English keywords can enhance clustering effectiveness. The lower performance of some combinations, however, may indicate that not all keyword sets are equally beneficial.

In our endeavor to attain optimal outcomes across diverse object representations, we employed ensemble clustering with two different algorithms: Agglomerative Clustering and Gaussian Mixture Models (GMM). These algorithms were selectively applied to the top-performing results derived from **Tables 4.5 & 4.6**. The parameters employed for these ensembles correspond to those used in the English data

Table 4.7: Ensemble clustering using Agglomerative and GMM in Arabic data with Arabic keyword

Technique	Metrics	GMM	Agglomerative
K1	S	0.1593421	0.71850187
	C	1649.8963204 892182	21486.4005414 01115
	D	2.32765516 48394856	0.475001023 5891053
D+K1	S	0.32002914	0.61412066
	C	1844.61684 33260316	19111.3507 40051
	D	1.15805099 47694495	0.5001732099 351155
M+K3	S	0.51984626	0.643016
	C	6740.43466 9158925	22072.439147 806228
	D	1.120529818 7833865	0.537266606 2900184
MD+K1	S	0.32002908	0.6141208
	C	1844.616534 5116276	19111.37097 666249
	D	1.158051340 6686361	0.50017325 84638654

The results indicate that Agglomerative clustering consistently outperforms GMM. The best result is achieved with the K1 technique using Agglomerative clustering, which has a Silhouette score of 0.71850187, indicating the most well-defined clusters among all tested methods.

Table 4.8: Ensemble clustering using Agglomerative and GMM in Arabic data with English keyword

Technique	Metrics	GMM	Agglomerative
K2	S	0.121401384	0.6966918
	C	1512.44895 4421008	22561.53657 427041
	D	1.82393898 13708986	0.56836458 83259612
D+K1	S	0.32034412	0.64278775
	C	3000.35720 189152	19793.23063 5288
	D	0.73496091 80878969	0.4708697 88169374
M+K1	S	0.30612046	0.7281388
	C	3149.410853 934293	22183.87983 856446
	D	0.88227760 13699509	0.453750951 0534045
MD+K1	S	0.32034364	0.6427878
	C	3000.35682 2271811	19793.247977 37653
	D	0.734960899 5940897	0.4708696 37734259

It is evident that Agglomerative clustering consistently outperforms GMM across various techniques. The highest silhouette score is achieved by the M+K1 technique using Agglomerative clustering, with a score of 0.7281388, indicating the most well-defined clusters among all tested methods.

2. Results of different clustering algorithms : in Tables 4.9 & 4.10, we showcase the outcomes achieved by employing diverse clustering algorithms with predefined parameters to generate ensemble clusterings. These algorithms were meticulously selected and applied to the highest-performing results obtained from different object representation methods. The parameters employed for these ensembles correspond to those used in the English data:

Table 4.9: Result of Ensemble clustering in Arabic data with Arabic keyword

Technique	Metrics	Ensemble of 3 Algorithmes	K-means+ Agglomerative	KMeans+ Gaussian	Gaussian+ Agglomerative
K1	S	0.7000599	0.71850175	0.6988605	0.71850175
	C	22634.541 956694975	21486.3426 81856633	22778.816 289394357	21486.3463 21423087
	D	0.55264525 47771405	0.47500135 258218773	0.55437818 92085458	0.47500127 521381236
D+K1	S	0.6141204	0.6141204	0.5625723	0.61412054
	C	19111.330 90167005	19111.350 15601468	11932.795 772317013	19111.340 254235278
	D	0.5001736 013155684	0.5002173 1919170821	0.67393076 57335979	0.5001732 19671117
M+K3	S	0.64693993	0.6430161	0.64713469	0.6430161
	C	18548.384 86767511	22072.408 504786086	16795.0442 43408553	22072.4180 86777285
	D	0.4707325 685474862	0.5372676 018291398	0.61776993 92118431	0.53726727 03493251
MD+K1	S	0.64693993	0.6141205	0.5625723	0.6141205
	C	18548.3848 6767511	19111.363 299672186	11932.795 14776359	19111.3571 2310097
	D	0.47073256 85474862	0.5001731 343586814	0.6739295 051639235	0.50017318 20524734

The analysis of Table 4.9 shows that the results of the algorithms are very close, with "Gaussian + Agglomerative" and "K-means + Agglomerative" emerging as the best techniques based on the K1 combination. Both achieve the highest Silhouette Score 0.71850175 . Although "Gaussian + Agglomerative" has a slightly higher Calinski-Harabasz Index (21486.3463), the overall performance of both methods is nearly identical.

Table 4.10: Result of Ensemble clustering in Arabic data with English keyword

Technique	Metrics	Ensemble of 3 Algorithmes	K-means+ Agglomerative	KMeans+ Gaussian	Gaussian+ Agglomerative
K2	S	0.42500076	0.6966917	0.6136196	0.69669193
	C	1815.15875 20773942	22561.5158 3453805	13839.6894 06508876	22561.527 367271476
	D	0.904147514 0237454	0.56836467 7117865	0.85503595 41722002	0.5683641 736965692
D+K1	S	0.6427876	0.64278764	0.38666186	0.64278775
	C	19793.236 50270161	19793.2452 02381346	3378.2104 74836714	19793.2442 85407935
	D	0.4708698 4732952536	0.47086966 91873984	2.1266757 317368796	0.47086964 819001986
M+K1	S	0.41997117	0.72813874	0.6819453	0.72813886
	C	1307.13965 9600541	22183.8646 83043903	17757.426 513964136	022183.8762 1494166
	D	1.03423398 13020565	0.45375107 78891658	0.5445479 014871696	0.45375091 599260187
MD+K1	S	0.64278746	0.6427875	0.38666183	0.64278764
	C	19793.2346 49712147	19793.2256 61602108	3378.2095 862802757	19793.2420 97297312
	D	0.47086981 91978227	0.47086987 73681029	2.1266754 889702426	0.47086959 32453015

Table 4.10 demonstrates that the "Gaussian + Agglomerative" algorithm consistently delivers the best performance in clustering Arabic data with English keywords. The stand-out result is a Silhouette Score of 0.72813886 for the M+K1 combination. This method also performs strongly across other combinations, as seen with high scores in the K2 and MD+K1 combinations.

3. Results of different parameter initialization: In Tables 4.11 & 4.12, we delve into the results obtained by employing various parameter initializations in ensemble clustering. These approaches were specifically applied to the method yielding the best results among different clustering algorithms. Furthermore, we introduced additional methods to augment our analysis. The parameters for these ensemble methods remained constant, with **clusters=6** and **n splits=22**. The subsequent combinations were utilized:

- **Ensemble 2-GMM and KMeans:**

GMM (n components=6, covariance type='tied')

KMeans (n clusters=6, init='k-means++', random state=42)

GMM (n components=6, covariance type='diag')

- **Ensemble 2-GMM and agglomeratives:**

GMM (n components=6, covariance type='tied')

Agglomerative (n components=6, linkage='ward')

GMM (n components=6, covariance type='diag')

Table 4.11: Results of different parameter initialization in Arabic data with Arabic keyword

Technique		2 GMM+k-means	2 GMM+ Agglomerative
K1	S	0.7002942	0.7000601
	C	22864.951720807945	22634.575392025527
	D	0.5525221284687479	0.5526448834732988
D+K1	S	0.5155605	0.61412084
	C	14499.509122854779	19111.371331202252
	D	0.9372483209107187	0.5001729691597286
M+K3	S	0.6647946	0.643016
	C	16837.816312943927	22072.420237907576
	D	0.5677052990982532	0.5372681517007785
MD+K1	S	0.5155602	0.6141208
	C	14499.515255871494	19111.362168928397
	D	0.9372486366531557	0.5001730304804352

Table 4.11 of different parameter initialization in Arabic data with Arabic keyword indicates that no single algorithm consistently outperforms the others across all techniques. Each algorithm demonstrates its strengths in different combinations. For instance, "2 GMM + k-means" shows its effectiveness with the highest Silhouette Score of 0.7002942 in the K1 combination, while "2 GMM + Agglomerative" performs well in other settings, such as achieving a Silhouette Score of 0.61412084 in the D+K1 combination. The highest result appears in the "2 GMM + k-means" with the M+K3 combination, reaching a Silhouette Score of 0.6647946.

Table 4.12: Results of different parameter initialization in Arabic data with English keyword

Technique		2 GMM+k-means	2 GMM+ Agglomerative
K2	S	0.6478891	0.42500088
	C	14405.17704586264	1815.1586911938282
	D	0.659988799074376	0.9041469249708827
D+K1	S	0.4430597	0.6427879
	C	8323.955095789228	19793.24324587737
	D	1.4353997372989493	0.47086961084815887
M+K1	S	0.6858503	0.42923573
	C	18595.923876520785	1311.265848280646
	D	0.5596238566122017	0.7645868761976568
MD+K1	S	0.44305965	0.64278764
	C	8323.958736914657	19793.24298049572
	D	1.4353999018288295	0.4708698353955045

Table 4.12 indicates that no single algorithm consistently outperforms the others across all techniques. Each algorithm demonstrates its strengths in different combinations. For instance, "2 GMM + k-means" shows its effectiveness with the highest Silhouette Score of 0.6858503 in the M+K1 combination, while "2 GMM + Agglomerative" performs well in other settings, such as achieving a Silhouette Score of 0.6427879 in the D+K1 combination. The highest result appears in the "2 GMM + k-means" with the K2 combination, reaching a Silhouette Score of 0.6478891.

The best results, based on the silhouette score, were achieved using English data and the Algerian Arabic dialect data, as illustrated in the graphical representation below:

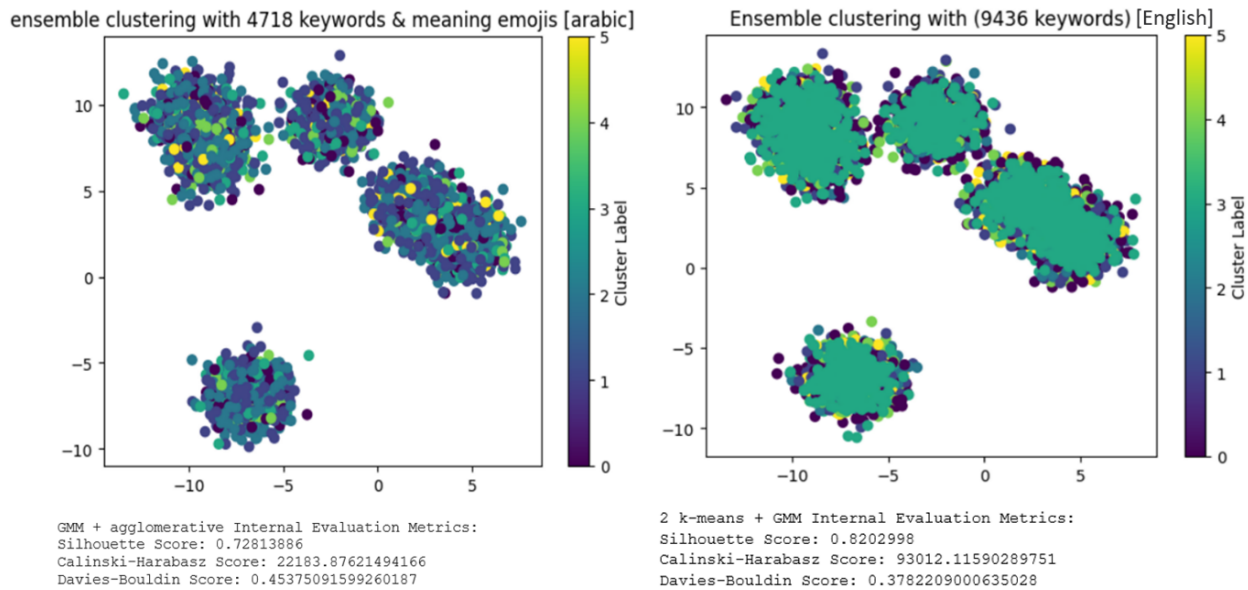


Figure 4.25: Best Result of Ensemble clustering

3.7 Conclusion

In this chapter we explored our methodology for detecting emotions from tweets, focusing on both Algerian dialectical and English tweets. We tried to enhance existing techniques by improving ensemble clustering methods. The chapter covered data collection, preprocessing, and the application of advanced machine learning models, particularly BERT, to achieve accurate emotion detection. Additionally, we presented the results and discussed their implications, providing a comprehensive overview of our findings and their significance.

In the next section we will display our final observations and perspectives in the context of the final conclusion.

General Conclusion

This study contributes to understanding how ensemble clustering techniques can improve the accuracy and robustness of emotion detection, addressing challenges in text analysis across diverse languages. The generation step, involving different object representation, different algorithms, and different clustering initialization, is crucial for these improvements. We specifically answered two key questions: Is ensemble clustering that combines multiple algorithms more effective than ensemble clustering using a single algorithm? And, which specific combination of ensemble clustering techniques yields the best results? Of course in the context of Emotion Detection from text.

In our work, we confronted the challenge of emotion detection by handling emojis and keywords with different methods, recognizing their significant role in conveying emotions on social media. We experimented with various combinations of clustering algorithms to achieve optimal results and modified the initialization of these algorithms to further refine the outcomes. By systematically testing and improving these approaches, we aimed to determine the most effective strategies for emotion detection, ultimately achieving the best results through our comprehensive methodology.

The results of this study show a notable improvement over previous work, with a significant increase in the accuracy of emotion detection. Specifically, the Arabic data achieved a higher performance with an ensemble of GMM and Agglomerative clustering. In English data, the best result was obtained using a combination of two K-means and GMM in the keyword-based analysis. One of the key limitations of our research is the inherent complexity of analyzing the Algerian dialect due to its multilingual nature, making it challenging to process and accurately detect emotions.

Extending this work could involve further refinement of the generation mechanisms, such as exploring alternative subspace projection techniques and example subsets. Additionally, future research could experiment with different algorithms, combinations and

initialization of algorithms to further improve emotion detection accuracy. Another potential avenue for enhancement is the selection of a consensus function other than simple voting, which could lead to more robust and reliable clustering outcomes. These extensions would help in addressing current limitations and advancing the field of emotion detection in multilingual and diverse text data.

Bibliography

1. Abdala, D. D. Ensemble and constrained clustering with applications (2010).
2. Abdallah, M. *PyArabic* <https://pypi.org/project/PyArabic/>. Latest version: June 18, 2022. 2022. (2024).
3. Abdul-Mageed, M. & Ungar, L. *Emonet: Fine-grained emotion detection with gated recurrent neural networks in Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (2017), 718–728.
4. Acheampong, F. A., Wenyu, C. & Nunoo-Mensah, H. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports* **2**, e12189 (2020).
5. Aggarwal, C. C. & Aggarwal, C. C. *Machine learning for text: An introduction* (Springer, 2018).
6. Agrawal, A. & An, A. *Unsupervised emotion detection from text using semantic and syntactic relations in 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* **1** (2012), 346–353.
7. Ahmed, M. H., Tiun, S., Omar, N. & Sani, N. S. Short text clustering algorithms, application and challenges: A survey. *Applied Sciences* **13**, 342 (2022).
8. Alqurashi, T. & Wang, W. Clustering ensemble method. *International Journal of Machine Learning and Cybernetics* **10**, 1227–1246 (2019).
9. Bandhakavi, A., Wiratunga, N., Massie, S. & Padmanabhan, D. Lexicon generation for emotion detection from text. *IEEE intelligent systems* **32**, 102–108 (2017).
10. Barnett, M. *regex* Latest version: May 15, 2024. 2024. <https://pypi.org/project/regex/> (2024).
11. Bengio, Y., Goodfellow, I. & Courville, A. *Deep learning* (MIT press Cambridge, MA, USA, 2017).

12. Berkhin, P. in *Grouping multidimensional data: Recent advances in clustering* 25–71 (Springer, 2006).
 13. Bird, Steven and Loper, Edward and Klein, Ewan. *Natural Language Toolkit (NLTK)* <https://www.nltk.org/>. Accessed: 2024-05-19. 2024.
 14. Bock, H.-H. Clustering methods: a history of k-means algorithms. *Selected contributions in data analysis and classification*, 161–172 (2007).
 15. Breiman, L. Bagging predictors. *Machine learning* **24**, 123–140 (1996).
 16. Béjar, J. *Unsupervised Machine Learning* Master in Artificial Intelligence - URL. 2022.
 17. Cesar, L. B., Manso-Callejo, M.-Á. & Cira, C.-I. BERT (Bidirectional Encoder Representations from Transformers) for missing data imputation in solar irradiance time series. *Engineering Proceedings* **39**, 26 (2023).
 18. Chakraborty, S. & Nagwani, N. K. Analysis and study of Incremental DBSCAN clustering algorithm. *arXiv preprint arXiv:1406.4754* (2014).
 19. Chen, Z. L. Research and application of clustering algorithm for text big data. *Computational Intelligence and Neuroscience* **2022**, 7042778 (2022).
 20. Chirra, V. R. R., Uyyala, S. R. & Kolli, V. K. K. Virtual facial expression recognition using deep CNN with ensemble learning. *Journal of Ambient Intelligence and Humanized Computing* **12**, 10581–10599 (2021).
 21. De Choudhury, M., Kiciman, E., Dredze, M., Coppersmith, G. & Kumar, M. *Discovering shifts to suicidal ideation from mental health content in social media* in *Proceedings of the 2016 CHI conference on human factors in computing systems* (2016), 2098–2110.
 22. Deepa, M. D. *et al.* Bidirectional encoder representations from transformers (BERT) language model for sentiment analysis task. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* **12**, 1708–1721 (2021).
 23. Desmet, B. & Hoste, V. Emotion detection in suicide notes. *Expert Systems with Applications* **40**, 6351–6358 (2013).
 24. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
-

25. Dhage, S. N. & Raina, C. K. A review on Machine Learning Techniques. *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)* **4**, 395–399. ISSN: 2321-8169 (2016).
 26. Dhall, A., Goecke, R., Lucey, S., Gedeon, T., *et al.* Collecting large, richly annotated facial-expression databases from movies. *IEEE multimedia* **19**, 34 (2012).
 27. Dridi, S. Supervised learning-a systematic literature review. *preprint, Dec* (2021).
 28. Ekman, P. Are there basic emotions? (1992).
 29. El Ayadi, M., Kamel, M. S. & Karray, F. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern recognition* **44**, 572–587 (2011).
 30. Emojipedia. *Smileys & Emotion* <https://emojipedia.org/smileys/>. Accessed: 2024-05-20. 2024.
 31. Face, H. *Hugging Face: The AI community building the future* 2024. <https://huggingface.co/> (2024).
 32. Facebook. *Facebook* <https://www.facebook.com/>. Accessed: 2024-06-01.
 33. Fan, Y., Lam, J. C. & Li, V. O. *Multi-region ensemble convolutional neural network for facial expression recognition in Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I* **27** (2018), 84–94.
 34. Fred, A. L. & Jain, A. K. Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence* **27**, 835–850 (2005).
 35. Freund, Y. & Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* **55**, 119–139. ISSN: 0022-0000. <https://www.sciencedirect.com/science/article/pii/S002200009791504X> (1997).
 36. Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232 (2001).
 37. Gan, Y., Chen, J. & Xu, L. Facial expression recognition boosted by soft label with a diverse ensemble. *Pattern Recognition Letters* **125**, 105–112 (2019).
-

38. Giachanou, A. & Crestani, F. Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)* **49**, 1–41 (2016).
 39. Google Colab. *Google Colaboratory: Welcome to Colaboratory* <https://colab.research.google.com/notebooks/intro.ipynb>. Accessed: 2024-05-19. 2024.
 40. Harris, C. R. *et al.* *Array programming with NumPy* <https://numpy.org/>. Accessed: 2024-05-19. 2020.
 41. Hawk, S. T., Fischer, A. H. & Van Kleef, G. A. Face the noise: embodied responses to nonverbal vocalizations of discrete emotions. *Journal of personality and social psychology* **102**, 796 (2012).
 42. HINTON, S. & Hjorth, L. *Understanding Social Media* 1st ed. English. ISBN: 9781446201206 (SAGE Publications Ltd, United Kingdom, 2013).
 43. Instagram. *Instagram: Social Media Platform* Accessed: 2024-05-31. 2024. <https://www.instagram.com>.
 44. Ira, N. T. & Rahman, M. O. *An efficient speech emotion recognition using ensemble method of supervised classifiers in 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)* (2020), 1–5.
 45. Ishengoma, F. R. in *Handbook of Research on Applications of AI, Digital Twin, and Internet of Things for Sustainable Development* chap. 26 (IGI Global, 2023). ISBN: 978-1-6684-6821-0. <https://www.igi-global.com/chapter/taxonomy-of-ethical-dilemmas-in-artificial-intelligence/>.
 46. Jain, K. & Kaushal, S. *A comparative study of machine learning and deep learning techniques for sentiment analysis in 2018 7th International conference on reliability, info-com technologies and optimization (Trends and Future Directions)(ICRITO)* (2018), 483–487.
 47. Jerritta, S, Murugappan, M, Nagarajan, R & Wan, K. *Physiological signals based human emotion recognition: a review in 2011 IEEE 7th international colloquium on signal processing and its applications* (2011), 410–415.
 48. Jing, L. *Survey of text clustering. Department of Mathematics, The University of Hong Kong, HongKong, China,, ISBN, 7695–1754* (2008).
-

49. John, O. P. & Gross, J. J. Healthy and unhealthy emotion regulation: Personality processes, individual differences, and life span development. *Journal of personality* **72**, 1301–1334 (2004).
 50. Kaggle. *Kaggle Search* <https://www.kaggle.com>. Accessed: 2024-06-01.
 51. Kameshwaran, K & Malarvizhi, K. Survey on clustering techniques in data mining. *International Journal of Computer Science and Information Technologies* **5**, 2272–2276 (2014).
 52. Kim, T. & Wurster, K. *emoji · PyPI* 2024. <https://pypi.org/project/emoji/>.
 53. Kruse, R., Döring, C. & Lesot, M.-J. Fundamentals of fuzzy clustering. *Advances in fuzzy clustering and its applications*, 3–30 (2007).
 54. Lazarus, R. S. Cognition and motivation in emotion. *American psychologist* **46**, 352 (1991).
 55. Levenson, R. W. The autonomic nervous system and emotion. *Emotion review* **6**, 100–112 (2014).
 56. Li, S., Deng, W. & Du, J. *Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 2852–2861.
 57. Lin, C.-R. & Chen, M.-S. Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging. *IEEE Transactions on Knowledge and Data Engineering* **17**, 145–159 (2005).
 58. Machová, K., Szabóová, M., Paralič, J. & Mičko, J. Detection of emotion by text analysis using machine learning. *Frontiers in Psychology* **14**, 1190326 (2023).
 59. Matplotlib Development Team. *Matplotlib: Visualization with Python* <https://matplotlib.org/>. Accessed: 2024-05-19. 2022.
 60. McKinney, W. *et al.* *pandas: powerful Python data analysis toolkit* <https://pandas.pydata.org/>. Accessed: 2024-05-19. 2022.
 61. Mekthanavanh, V., Li, T., Hu, J. & Yang, Y. *Web video clustering based on emotion category* in *Proceedings of the 2018 International Conference on Big Data Engineering and Technology* (2018), 87–91.
-

62. Meng, Q. *et al.* Unsupervised representation learning for time series: A review. *arXiv preprint arXiv:2308.01578* (2023).
 63. Metzler, H., Pellert, M. & Garcia, D. *Using Social Media Data to Capture Emotions Before and During COVID-19* 2022. <https://worldhappiness.report/ed/2022/using-social-media-data-to-capture-emotions-before-and-during-covid-19/>.
 64. MEZATI, M., Bougoffa, A. Z. A. & Khediri, A. *Text Clustering in Social Media* MA thesis (Kasdi Merbah University of OUARGLA, 2023).
 65. Mitchell, T. M. *Machine Learning* 432 (McGraw-Hill Science/Engineering/Math, 1997).
 66. Mohammad, S. M. & Bravo-Marquez, F. Emotion intensities in tweets. *arXiv preprint arXiv:1708.03696* (2017).
 67. Muhammad, I. & Yan, Z. SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. *ICTACT Journal on Soft Computing* **5** (2015).
 68. Munezero, M., Montero, C. S., Sutinen, E. & Pajunen, J. Are they different? Affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing* **5**, 101–111 (2014).
 69. Nandwani, P. & Verma, R. A review on sentiment analysis and emotion detection from text. *Social network analysis and mining* **11**, 81 (2021).
 70. Nomiya, H., Morikuni, A. & Hochin, T. *An Unsupervised Emotional Scene Retrieval Framework for Lifelog Videos in 2014 IIAI 3rd International Conference on Advanced Applied Informatics* (2014), 609–615.
 71. Ortony, A., Clore, G. L. & Collins, A. *The cognitive structure of emotions* (Cambridge university press, 2022).
 72. Pajupuu, H., Kerge, K. & Altrov, R. Lexicon-based detection of emotion in different types of texts: Preliminary remarks. *Eesti Rakenduslingvistika Ühingu aastaraamat* **8**, 171–184 (2012).
 73. Pedregosa, F. *et al.* *scikit-learn: Machine Learning in Python* <https://scikit-learn.org/stable/>. Accessed: 2024-05-19. 2024.
 74. Perikos, I. & Hatzilygeroudis, I. Recognizing emotions in text using ensemble of classifiers. *Engineering Applications of Artificial Intelligence* **51**, 191–201 (2016).
-

75. Picard, R. W. Affective computing: challenges. *International Journal of Human-Computer Studies* **59**, 55–64 (2003).
 76. Piliapp. *Emoji List* <https://www.piliapp.com/emoji/list/>. Accessed: 2024-05-19. 2024.
 77. Plutchik, R. in *Theories of emotion* 3–33 (Elsevier, 1980).
 78. Python Software Foundation. *Python: Programming Language* <https://www.python.org/>. Accessed: 2024-05-19. 2024.
 79. Rai, P. & Singh, S. A survey of clustering techniques. *International Journal of Computer Applications* **7**, 1–5 (2010).
 80. Razek, M. A. & Frasson, C. Text-based intelligent learning emotion system. *Journal of Intelligent Learning Systems and Applications* **9**, 17–20 (2017).
 81. Reddit. *Reddit: The Front Page of the Internet* Accessed: 2024-05-31. 2024. <https://www.reddit.com>.
 82. Reimers, N. & Aarsen, T. *Sentence Transformers* Latest version: Apr 17, 2024. 2024. <https://pypi.org/project/sentence-transformers/> (2024).
 83. Reimers, N. & Aarsen, T. *Transformers* Latest version: May 17, 2024. 2024. <https://pypi.org/project/transformers/> (2024).
 84. Roberts, K., Roach, M. A., Johnson, J., Guthrie, J. & Harabagiu, S. M. *EmpaTweet: Annotating and Detecting Emotions on Twitter*. in *Lrec* **12** (2012), 3806–3813.
 85. Rodriguez, J. J., Kuncheva, L. I. & Alonso, C. J. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence* **28**, 1619–1630 (2006).
 86. Rodriguez, P., Ortigosa, A. & Carro, R. M. *Extracting emotions from texts in e-learning environments* in *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems* (2012), 887–892.
 87. Russell, J. A. A circumplex model of affect. *Journal of personality and social psychology* **39**, 1161 (1980).
 88. Russell, J. A. & Mehrabian, A. Evidence for a three-factor theory of emotions. *Journal of research in Personality* **11**, 273–294 (1977).
-

89. Sagi, O. & Rokach, L. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery* **8**, e1249 (2018).
 90. Salam, S. A. & Gupta, R. Emotion detection and recognition from text using machine learning. *Int. J. Comput. Sci. Eng* **6**, 341–345 (2018).
 91. Sariyanidi, E., Gunes, H. & Cavallaro, A. Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE transactions on pattern analysis and machine intelligence* **37**, 1113–1133 (2014).
 92. Seyeditabari, A., Tabari, N. & Zadrozny, W. Emotion detection in text: a review. *arXiv preprint arXiv:1806.00674* (2018).
 93. Shen, J. *et al.* Real-time superpixel segmentation by DBSCAN clustering algorithm. *IEEE transactions on image processing* **25**, 5933–5942 (2016).
 94. Shih, P.-Y., Chen, C.-P. & Wu, C.-H. *Speech emotion recognition with ensemble learning methods in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), 2756–2760.
 95. Shrivastava, P. & Gupta, H. A review of density-based clustering in spatial data. *International Journal of Advanced Computer Research* **2**, 200 (2012).
 96. Siddiqui, T. & Aalam, P. Short Text Clustering; Challenges Solutions: A Literature Review (June 2015).
 97. Sinaga, K. P. & Yang, M.-S. Unsupervised K-means clustering algorithm. *IEEE access* **8**, 80716–80727 (2020).
 98. Topchy, A., Jain, A. K. & Punch, W. *A mixture model for clustering ensembles in Proceedings of the 2004 SIAM international conference on data mining* (2004), 379–390.
 99. Vega-Pons, S. & Ruiz-Shulcloper, J. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence* **25**, 337–372 (2011).
 100. Walde, S. Chapter 4: Clustering algorithms and evaluations. *Experiments on the automatic induction of german semantic verb classes* **9**, 179–205 (2003).
 101. Wang, G., Sun, J., Ma, J., Xu, K. & Gu, J. Sentiment classification: The contribution of ensemble learning. *Decision support systems* **57**, 77–93 (2014).
 102. Watson, D. S. On the philosophy of unsupervised learning. *Philosophy & Technology* **36**, 28 (2023).
-

103. X Corp. *Twitter API Documentation* <https://developer.x.com/en/docs/twitter-api>. Accessed: 2024-06-01.
 104. X Corp. *X* <https://x.com/?mx=2>. Accessed: 2024-06-01.
 105. Yang, S., Huang, G. & Cai, B. Discovering topic representative terms for short text clustering. *IEEE Access* **7**, 92037–92047 (2019).
 106. Yang, Y. & Jiang, J. Hybrid sampling-based clustering ensemble with global and local constitutions. *IEEE transactions on neural networks and learning systems* **27**, 952–965 (2015).
 107. Yuan, S., Huang, H. & Wu, L. Use of word clustering to improve emotion recognition from short text. *Journal of Computing Science and Engineering* **10**, 103–110 (2016).
 108. Zhang, L. & Liu, B. *Sentiment Analysis and Opinion Mining in Synthesis Lectures on Human Language Technologies* (2012).
 109. Zvarevashe, K. & Olugbara, O. Ensemble learning of hybrid acoustic features for speech emotion recognition. *Algorithms* **13**, 70 (2020).
-