الجامعة البريطانية في دبي

The British University in Dubai

Exploring Sentiment Analysis using Different Machine Learning Algorithms on Dialectal Arabic

دراسة تحليل المشاعرباستخدام خوارزميات التعلم الآلي المختلفة على اللهجات العربية

by

MOUZA MATAR AL MANSOORI

Dissertation submitted in fulfilment

of the requirement for the degree of

MSc INFORMATICS

at

The British University in Dubai

April 2021

DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

Signature of the student

COPYRIGHT AND INFORMATION TO USERS

Abstract

Today, the intense use and relaying on the modern technologies such as; mobile phones, e-commerce, interactive websites, social media, wearables technologies, sensors, and satellites, is enabling data to be generated every second resulting in huge structured and unstructured data availability. Therefore, big data analytics field emerged to tame the generated big data, and use it to provide useful insights to the world. Sentiment Analysis is one application of big data analysis when dealing with text data. Sentiment Analysis refers to the processes of extracting and analysing emotions from a given text to classify its polarity, mainly within three classes; positive, negative and neutral. Many researches have been done on Sentiment Analysis for English text data. While more exploration is still required to be done on Arabic twitter sentiment analysis. This paper focuses on dialectal Arabic sentiment analysis. The study explores sentiment analysis using different machine learning algorithms on dialectal Arabic text dataset. In this study, we used twitter as our data source. Therefore, our dataset consists of Arabic tweets. The purpose of this study is to examine the performance of sentiment analysis on three datasets that have different level of dialectal Arabic; mixed dialects dataset, gulf dialect dataset, and Emirati dialect dataset. Two machine learning classifiers were used in this experiment; the support vector machine (SVM) and Naïve Bayes (NB). The results of this experiments indicate that when applying sentiment analysis on one specific dialect group, the performance accuracy is higher than the performance of sentiment analysis on the mix dialects dataset under same settings. The experiment also supports other studies in that the SVM classifier outperformed NB classifier. We conclude that additional research is required to be done to explore more on Arabic sentiment analysis considering different dialects.

الملخص (Abstract)

الاعتماد الكبير اليوم على استخدام التكنولوجيا الحديثة - من مثل الهواتف المتحركة، التجارة الإلكترونية، المواقع التفاعلية ومواقع التواصل الاجتماعي، الأدوات التقنية القابلة للارتداء، أجهزة الاستشعار عن بعد والأقمار الصناعية – مكّنت عملية تكوّن البيانات بسرعه فائقة، حيث يتم انتاج بيانات جديدة كل ثانية بسبب استخدام هذه التقنيات، مما أدى إلى توفّر بيانات ضخمة. وللتعامل مع هذه البيانات ظهر مجال "تحليل البيانات الكبيرة"، حيث يتم تحليل البيانات الكبيرة واستخدامها لاستنباط استدلالات مفيدة عن العالم. إن تقنية "تحليل المشاعر" هي أحد مجالات "تحليل البيانات الكبيرة". تحليل المشاعر هي تقنية يتم من خلالها تحديد المشاعر والآراء من البيانات النصية وتحليلها لتصنيف اتجاهها، حيث يتم تصنيف الاتجاه بشكل رئيسي ضمن ثلاث تصنيفات: إيجابي، سلبي، ومحايد. تم إعداد العديد من الأبحاث حول تحليل المشاعر للبيانات النصية باللغة الإنجليزية. بينما لا تزال البحوث على البيانات النصية باللغة العربية قليلة وتتطلب المزيد من الاستكشاف. هذا البحث يدرس تطبيق تحليل المشاعر – باستخدام خوارزميات التعلم الآلي - على اللهجات العربية. تم استخدام بيانات تويتر كمصر للبيانات في هذا البحث. الهدف من هذا البحث هو دراسة تقنية تحليل المشاعر على ثلاث قواعد بيانات لمستويات مختلفة من اللهجات العربية، قاعدة البيانات الأولى تحتوى على تغريدات بلهجات عربية مختلفة، قاعد البيانات الثانية تحتوى على تغريدات باللهجة الخليجية، وقاعدة البيانات الثالثة تحتوى على تغريدات باللهجة الإماراتية. تم تجربة تقنيتين من تقنيات التعلم الآلي، الأولى هي "آلة المتجه الداعم (SVM)" ، والثانية هي " المصنف البايزي الساذج (NB)”. تشير نتائج تجارب هذا البحث إلى أنه عند تطبيق تحليل المشاعر على مجموعة البيانات التي تحتوي على مجموعة من اللهجات المتشابهه (مثل اللهجات الخليجية)، تكون النتيجة أكثر دقة من تطبيقها على مجموعة البيانات التي تحتوي على اللهجات المختلطة تحت نفس الظروف. تدعم التجربة أيضًا دراسات أخرى التي تستنتج أن الخوارزمية SVM تتفوق في الأداء على الخوارزمية NB. ونوصي بإعداد دراسات إضافية لدراسة تطبيق تقنية تحليل المشاعر على النصوص العربية، وتحديداً اللهجات العربية المختلفة.

Acknowledgements

I am Thankful for the opportunity the British University in Dubai (BUiD) provided me to write this experimental research project that explores “Dialectal Arabic sentiment analysis”. I also offer my appreciation to the BUiD administrators, librarians, and technical supporters.

I am also sincerely thankful to Dr.Khaled Shaalan for his support and guidance throughout this project. Without his encouragement and support it would be difficult to achieve such a high standard research paper.

Finally, I would like to acknowledge my professors of “Informatics” major who provided me with valuable knowledge and insights; Dr.Khaled Shaalan, Dr. Sherief Abdallah, and Dr. Cornelius Ncube.

Table of Contents

List of Illustrations

List of Tables

List of definitions and abbreviations

#	Term	Abbreviation	Definition
1	Arabizi	-	It refers to when Arabic word is written using Latin characters
2	Computational Linguistics	CL	Subfield of Artificial Intelligence, emerged to deal with unstructured text data (natural human language). Also called Natural Language Processing.
3	Data Cleansing	-	The process of detecting missing, incorrect, or irrelevant data in a dataset, and treating it by deleting, correcting, or assigning it a value.
4	Deep Learning	DL	Subfield of Machine Learning, based on Artificial Neural Networks with multiple layers and feature learning. It works well with complex data and can achieve high accuracy results.
5	Feature Extraction	-	The process of dimensionality reduction, as it attempts to reduce the number of features in the datasets. It first create new features from the original features, then discard the old ones and use the new ones.
6	Gated Recurrent Unit	GRU	Deep learning method used to enhance the Recurrent Neural Networks (RNNs). It is similar to Long Short-Term Memory (LSTM) with different inner work, and has two gates while LSTM has three gates.
7	Long short-term memory	LSTM	Deep learning model that occurs to enhance the Recurrent Neural networks (RNNs). Instead of feedforward, it feedbacks the connections and process the entire data sequences. So LSTM can hold information in memory longer than RNNs.
8	Naïve Bayes	NB	Probabilistic Machine Learning model (classifier) used to achieve classification tasks based on the Bayes theorem. It assumes independencies of the pair of features - being classified - from other predictors.
9	Natural Language Processing	NLP	Subfield of Artificial Intelligence, emerged to deal with unstructured text data (natural human language). NLP also called Computational Linguistic (CL).
10	Sentiment Analysis	SA	The process of extracting useful insights on people’s emotions and opinions toward any topic such as entities, events, products, or services
11	stemming		The process of eliminating features where each word is shorten to its stem (root).
12	Support Vector Machine	SVM	Supervised Machine-learning model used for classification or regression tasks. It is based on finding a hyperplane in N dimension space, where N is the numer of features. SVM can create linear classification and non-linear classification.
13	Term Frequency — Inverse Document Frequency	TF-IDF	A technique to count words in a document, by calculating weight to define the significance of each word (Feature extraction method)
14	Tokenization		The process of splitting given sentence in to smaller segments (tokens).

1 Introduction

The enormous growth of data and the popularity of the modern technologies of Artificial Intelligence - today - emerged great interest in big data analytics, coveting to extract valuable insights for better understanding and better decisions making. Social media is one great source of big data, where content is immensely generated every second by internet users. Social media data is highly unstructured text data, that contains text, characters, numbers, URLs, emoji, pictures, media, symbols, hashtags, and mentions.

Natural Language Processing (NLP) field has emerged to deal with the unstructured text data, in other words, it attempts to understand and deal with natural human language (Zahidi, Younoussi, & Al-Amrani 2021). NLP field is an intersection field of; computer science, data science, artificial intelligence, text mining, social science and linguistics. NLP aims to automate some tasks such as question answering and language translation. Besides, there are many applications of the NLP, such as speech recognition, information retrieval, machine translation, chatbot, as well as sentiment analysis. NLP is also called Computational Linguistics (CL).

Sentiment Analysis is one of the most promising instruments for public opinion monitoring and measurement of the text data from social media platforms. Sentiment analysis is one of the most dynamic NLP fields, and it can be defined as the process of extracting useful insights on people's emotions and opinions toward any topic such as entities, events, products, or services (Liu 2012). Therefore, sentiment analysis is also called by the term opinion mining. The main objective of sentiment analysis is to identify the polarity of the text under discussion, where the following categories are generally identified: positive, negative, or neutral. The classification can be attained using Machine Learning approaches and/or lexicon-based approaches.

Sentiment analysis can be performed - mainly - at three levels; document level, sentence-level, and subject (or aspect) level (Boudad et al. 2018). At document-level, the classification is performed on the entire given text. The entire document is given one class, such as positive, negative or neutral. While, at sentence-level, the classification is performed on a single sentence, where each sentence could have different sentiment than the other sentence, in a given text (Nejjari & Meziane 2019). Furthermore, at aspect level, the classification is done based on specific features of the object that is extracted or identified. There are two tasks in aspect level; first is to extract aspects (attributes), second is to classify the sentiment of the different extracted aspects of an object. This means each aspect of an object may have different sentiment. (Boudad et al. 2018; Areed et al. 2020)

Sentiment analysis can be extremely useful as it provides useful insights on public overall opinions. It is important for governments and businesses to analyse people's emotions because people make decisions and act based on their emotions. Many different fields can benefit from sentiment analysis in different aspects such as releasing new product, service or brand, customers satisfaction, e-commerce, government, public health, tourism, politics, and education (Baali & Ghneim 2019; Boudad et al. 2018; Almuqren, Qasem & Cristea 2019).

Sentiment Analysis can be an effective tool to monitor and measure customer satisfaction. The availability of live large data and capability to automatically classify customer satisfaction will only improve businesses reputation as they can take better decision and control situations towards increasing their customer satisfaction and staying ahead of competitors. For example, Kuman and Zymbler (2019) are interested in implementing sentiment analysis to measure and analyze customer satisfaction toward airline services. Another example is the application of sentiment analysis to measure and predict customer satisfaction on telecommunication companies (Almuqren, Qasem & Cristea 2019).

Business decision makers and marketers can also utilize sentiment analysis to understand the public needs, market's trends, users' emotions and opinions regarding the under-discussion services, products and brands. Therefore, they can ensure to align and adjust their business and/or marketing strategy accordingly. They can also take decisions regarding launching new businesses, services, and/or products. Similarly, government, as important decision maker, may use sentiment analysis to understand citizen's overall opinion, take better decisions, and enhance the provided services (Nejjari & Meziane 2019). In the health field, sentiment analysis can be used to observe public and individual's opinion towards treatment, health care facilities, public health…etc (Alayba et al. 2017). One more relevant example today is monitoring public attitude toward the event of COVID-19 pandemic (Manguri, Ramadhan & Amin 2020). Furthermore, in political field, sentiment analysis can help in monitoring and understanding public – or group – attitudes regarding recent political related events, which could assist government to take better decisions and control circumstances.

In this paper, we are discussing and exploring the technicality of sentiment analysis, because sentiment analysis is very important today. It's important for the government, decision makers, business leaders, and researchers to understand how people feel and think toward a given subject. Only through better understanding, better actions can be taken. The development of the modern business depends on the effectiveness of customers' opinions analysis. To reach a better understanding of the potential audience's characteristics, an analyst must be able to analyze a large relevant data that would characterize customers' attitudes and beliefs expressed freely. In fact, there is a growing interest on sentiment analysis, and it is becoming a widely popular research field today (Ghallab, Mohsen & Ali 2020; Nejjari & Meziane 2019; Salloum et al. 2018).

1.1 Background Information

Nowadays, people widely use social media platforms such as Twitter, Facebook, Instagram, LinkedIn, and Goodreads, to easily communicate and freely express their personal thoughts, opinions, and emotions towards any subjects. For this reasons, social media content is great data source for sentiment analysis applications. One of the most used social media worldwide is Twitter, with more than 353 Millions of active users who use twitter to express their feelings and opinions (Tankovska 2021). For twitter sentiment analysis, language is important as we deal with text data.

Arab world is interesting, considering the rapid growth and dynamic changes in both social and economic sectors, strategic location with its historical records, political events, and the high potential of investment opportunities. Arabs speak and communicate in Arabic language.

1.1.1 Arabic Language

Arabic language is one of the most important languages. It is ranked as the fifth most spoken languages in the world, with 274 million speakers (Szmigiera 2021). Arabic language is the official language in 26 countries (Wikipedia 2021), and according to United Nation (UN) official website, Arabic language is one of their six official languages.

There are three major types of Arabic languages; Classic Arabic, Modern Standard Arabic (MSA), and dialectal Arabic (Nejjari & Meziane 2019; Shaalan et al. 2019; Hegazi et al. 2021; DoniaGamal et al. 2018). The Classic Arabic is found in some religious scrips, and not used in today's official and unofficial communications. While the Modern Standard Arabic (MSA) is the current official language to communicate in Arabic. For example, books, magazines, official websites, instructions and manuals, as well as official news are written and spoken using the Modern Standard Arabic (MSA). Arabs in different regions are familiar with the Modern Standard Arabic (MSA). On the other hand, dialectal Arabic language is different in different regions. Arabic dialects can be categorized into five major groups; Gulf, Egyptian, Levantine, Iraqi and Maghrebi (Soufan 2019). This categorization is based on geographical location, where each dialect group is spoken by countries that are near to each other. However, it is important to mention that one dialect group has many dialects that might sound similar but actually different. For example, the dialect of Emirati is different from Saudi dialect. Table 1 demonstrate dialect groups with its regions.

Arabic Dialects group	Spoken by
Gulf	UAE, Saudi Arabia, Kuwait, Bahrain, Qatar, Oman and Yemen
Egyptian	Egypt and Sudan
Levantine	Levant region (Jordan, Lebanon, Palestine and Syria)
Iraqi	Iraq
Maghrebi	North Africa countries; Tunisia, Morocco, Algeria, Libya, and Mauritania

***Table 1**: Arabic Dialect Groups*

Considering the importance of Arabic language, there is a high interest in the study of sentiment analysis methods for social media text data in Arabic language. This interest is a result of the large number of Arabic speakers in the world, and also because of the active Arab Internet users continuously posting, sharing and interacting on social media. Therefore, the Arabic content on the internet is continuously increasing. In fact, Arabic is the fourth most common language used on the Internet (Johnson 2021), and twitter is one of the most used social networking platforms among Arabs (DoniaGamal et al. 2018). This makes twitter one perfect source of data for Arabic sentiment analysis researches. Ghallab, Mohsen and Ali (2020) surveyed Arabic sentiment analysis systematically, and stated that most researches used twitter as their data source. However, Arabic sentiment analysis is a challenging task due to 1) the language complexity, and 2) the unstructured nature of the social media text data.

1.1.2 Challenges with Arabic Language Sentiment Analysis

Many researchers discussed various challenges associated with Arabic sentiment analysis task (Oueslati et al 2020; Nejjari & Meziane 2019; Shaalan et al. 2019; Zahidi, Younoussi, & Al-Amrani 2021; Ghallab, Mohsen & Ali 2020; Boudad et al. 2018). Most of these challenges are related to the complexity of the Arabic language nature. This section discusses the main challenges in Arabic sentiment analysis.

1. The complexity of the Arabic language morphology.

   Arabic language has rich and complex structure and morphology. While the sentiment analysis performance mainly depends on the morphology of the language, the structure of the Arabic language and its morphology makes sentiment Analysis a complex task. One interesting example is the root (كتب), which is same for many unrelated derivation words such as (books - كُتب), (office - مكتب), and (write – يكتب). Furthermore, in some cases it is difficult to distinguish between a word and a sentence. For example, (وسيعلمون) is a sentence that looks like a word. It means “and they will know”.

2. The flexibility in the free sentence order

   Another reason why Arabic language is challenging is the free sentence order. The same sentence in Arabic language can be written either as nominal sentence (starts with subject, followed by verb) or as verbal sentence (starts with verb, followed by subject). This is different in comparison with English language.

3. No capitalization in Arabic language

   Another way Arabic text is different in that there is no capitalization in Arabic language. This causes some confusion for the machine to separate the names from other words, especially when two words are written exactly similar. For example, the word أمل is a popular girl name, but it is a widely used sentiment word that means “hope” (Shaalan et al. 2019). Another example is the boy’s name سعيد, which is also a widely used sentiment word that means “happy”. The reason for this is that many Arabs prefer to use positive adjectives as names (Boudad et al. 2018)

4. The variations of the letter shape

   The letters in the Arabic language change its shape depending on its placement in the word. For example, this is how the letter م is written; in the beginning of the word (مـ), in middle of the word (ـمـ), and at the end of the word (ـم ، م).

5. The use – and the absence– of the diacritic Marks
   Diacritic marks are used above or below the letter, presenting a short vowel. It is mainly used to demonstrate the right pronunciations (Shaalan et al. 2019). The challenge with diacritics is that nowadays it is only used with classic Arabic and in children’s books. It is expected that Arabic speakers understand the pronunciations and the meanings without the use of diacritical marks. For this reason, the diacritics are optional in Arabic writing. The problem here is that several words would look the exact same without diacritics. One example is the word (شعر) which could mean poetry (شِعْرٌ), hair (شَعَر), or to feel (شَعَرَ) (Boudad et al. 2018).

6. Dialects of Arabic Language

   A wide variation of the Arabic dialects is the major challenge of Arabic sentiment analysis, and it is discussed by many researchers (Soufan 2019; Boudad et al. 2018; Al-Thubaity, Alqahtani & Aljandal 2018; Nejjari & Meziane 2019; Almuqren, Qasem & Cristea 2019). Each region in the Arab world has its own dialect. For example, in UAE the National population speaks the Emirati dialect. Internet Arab users mainly use their regional dialects when communicating through social media. Considering there are no standards for dialectal language structure nor for spelling, it's difficult to develop tools for Arabic dialects.

7. The use of Arabizi

   Arabizi is another layer of complexity when analyzing Arabic textual data (Boudad et al. 2018). Arabizi (Romanized Arabic) is the term used for writing Arabic word using Latin characters. For example, the word "شكراً", which means "thank you", is writen in Arabizi as "shokran". Arabizi started earlier when technology wasn't yet supporting typing in Arabic language. Gradually, it become a popular way of communication through internet by Arab users. In the sentiment analysis, the non-Arabic letters are usually removed in the phase of pre-processing, in which then we lose these messages written in Arabizi.

8. Bilingual Arabs using mix languages.

   Furthermore, Arabic speakers may mix two or three languages when texting or speaking, especially the younger generations. This is because they are bilingual or trilingual. Therefore, the tweet may be in Arabic with few foreign words. The foreign words usually written using Arabic characters. Furthermore, there are some words Arabs got from English where the Arabic word for it is not popular or doesn't exist. For example the following words (mention – منشن), (Block - بلوك), and (Retweet - ريتويت). These words are called "loanwords" (Alruily 2020).

9. Lack of tools and resources

   There have been many tools and resources developed to enhance the performances of sentiment analysis projects. However, these tools and resources doesn't mainly support Arabic sentiment analysis projects (Baali & Ghneim 2019; Soufan 2019; (Zahidi, Younoussi, & Al-Amrani 2021; Ghallab, Mohsen & Ali 2020) Furthermore, the tools that are developed to support Arabic are mainly working good for MSA, while it does not work well with dialectal Arabic. The lack of resources and tools for Arabic sentiment analysis makes it a challenging task for researchers. Most importantly, the developed tools for Arabic sentiment analysis tools by the recent researcher, do not get released for the other researchers. Baly et al. (2019) stated that the available resources are still basic, especially in the annotation process, in comparison to English recourses.

10. Sarcasm

Other issues with sentiment analysis in general are with Sarcasm (Siddiqui, Monem & Shaalan 2018). Sarcasm is an expression that may have hidden meaning, in which the meaning is not a straightforward. In most cases positive words are used to mean something negative in sarcasm. Table 2 presents some examples of sarcasm in tweets.

Arabic phrase	English translation	The actual meaning
باجر عندي امتحان و لي الحين ما 💔 ☺ درست عادي صح	Having an exam tomorrow, and I didn't study until now. Its ok isn't it.	I know its not ok that I didn't study yet for tomorrow's exam.
ترا صلة الرحم مالها شغل اني اقعد اضيفك بكل البرامج	Being relatives has nothing to do with following each other in all social media.	Even if we are relatives, I don't want to add you in all social media.
مب جنه وايد دخلتو ف حيات الولد😂 😂	Doesn't it seem like you are interfering a lot in his life.	Leave him along.
خلاااااص ترا بيحسبون الناس اني ادفع لك 🐦 💔 عموما الفلوس حولتها 👈 🐦	Enough, people will think that I'm paying you. Anyway, I transferred the money.	What you are say sounds too good.

***Table 2**: Sarcasm examples from twitter*

Additional challenges well discussed by Alruily (2020) who listed the following challenges in their research "shortening, compounded words, misspelled words, abbreviations, dialectal words (slang), neologisms, concatenation, word elongation and idiomatic expressions". An example of the compound words problem is a compound of the word (هذا) and the word (الدعم) into (هالدعم). Another widely observed word compound is the word (يا) and the words (صديقي), for example, into (ي صديقي). An example of the elongation is the word (الحياة) which becomes (الحيااااااة) after elongated. Beside Arabic language related challenges, there are other challenges related to using twitter platform. These challenges were discussed by many researches. For example, in the shortening problem, and because of the limited words allowed in a tweet – user, in some cases, try to find a way to shorten some words and phrases to fit more words in a tweet.

1.1.3 Sentiment Analysis Approaches

The sentiment analysis is, generally, performed through three main approaches; the lexicon-based approach, the machine learning approach, and the hybrid approach (Oueslati et al 2020; Abdulla et al. 2018). The approaches can also be categorized as supervised learning approaches, unsupervised learning approaches, and semi-supervised learning approaches. The main difference between the supervised learning and unsupervised learning is that large amount of labelled dataset is required in the supervised learning to train and build the classification model (Abo, Raj, & Qazi, 2019). In the next few paragraphs, a brief about these sentiment analysis approaches is presented.

Lexicon-Based approach is an unsupervised approach. it is an extension of the rules-based approaches where human define linguistic rules for the machine to extract keywords from a text (Al-Ayyoub et al. 2019). In lexicon-based approach a dictionary - or sentiment lexicon - is required. The dictionary contains sentiment related words with its sentiment class, usually either positive, negative or neutral. In its simplest forms, the system will identify the keywords in a given sentence, then it will look for them in the lexicon and will use the equivalent class to determine its polarity. If most words in the given sentence are positive, then the sentence sentiment is positive (Nejjari & Meziane 2019; Boudad et al. 2018). However, the first step here is to build sentiment lexicon which can be achieved through many different ways; manually by linguistic experts, automatically through word seeds frequency calculations, and semiautomatically constructed lexicon that required some tasks to be manually done (Areed et al. 2020; Al-Thubaity, Alqahtani & Aljandal 2018; Kaity & Balakrishnan 2019).

Machine Learning is an approach where the machine learns the rules from the use of training data. Large amount of labelled data is required to train the classifier. Therefore, the first step in the supervised machine learning approach is to split the dataset into training dataset and test dataset. Then, the training dataset is used to train the classifier and build the classification model. Next, the classifier will be used on a new dataset - the test dataset - to predict the class. After that, the performance of the classifier is evaluated using evaluation measurements such as accuracy and cross validation (Nejjari & Meziane 2019). Some of the most used Machine Learning algorithms in sentiment analysis are: Support Vector Machine (SVM), Regression, Random Forest (RF), Decision Tree (DT), Naïve Bayes (NB), and K-Nearest Neighbour

(KNN). Its important to mention the deep learning approach, which is an advance subfield of machine learning. Deep learning is newly currently used in many sentiment analysis researches (Al-Smadi et al. 2019). Some of the most used deep learning techniques in sentiment analysis are: Artificial Neural Networks (ANN) Convolutional neural networks CNNs, back-propagation Neural Networks BPNN, and Long short-term memory LSTM.

The hybrid approaches are also known as semi-supervised learning approaches. As indicated from its name, it combines and use both approaches to enhance the classifier performance. For this reason, both labelled dataset and unlabelled dataset are used in this approach. Basically, the classifier is trained using the labelled data, and then it is used on the unlabelled data to perform the classification.

1.2 Motivation

The motivation behind this paper is the importance of the Arabic sentiment analysis field itself. Our motivation was discussed earlier under the following logic;

1 The modern technologies encourage users to easily share their ideas, thoughts, and emotions on different social media platforms. Therefore, data is continuously generated in form of big data.

2 The availability of social media big data – along with NLP technologies – create an opportunity to analyze and learn from the data, using sentiment analysis.

3 The insights of social media big data can then be used to enhance current situations regarding any topic.

4 Arabic language is popular and widely used. Therefore, it’s important to utilized sentiment analysis tools to adopt Arabic language.

5 However, Arabic language is complex due to its structure, morphology, and variations. More efforts on Arabic language sentiment analysis are still required.

In addition, researches on sentiment analysis for Arabic language is still under progression, in comparison to the English sentiment analysis researches (Salloum et al. 2018; Al-Ayyoub et al. 2019). Furthermore, not enough researches have been done on dialectal Arabic. Based on the surveyed sample by Ghallab, Mohsen and Ali (2020), only around 21% of the researches are done of dialectal Arabic as independent from the Modern Standard Arabic (MSA).

Moreover, very limited researches have been done on the Gulf dialects in particular. In fact, among these researches, most work done toward Saudi dialect, and to our best knowledge, only one research focused on the Emirati Dialect sentiment analysis (Al Suwaidi, Soomro & Shaalan 2016).

1.3 Problem Statement

This paper explores the use of different machine learning algorithms on dialectal Arabic twitter sentiment analysis. We attempt to examine the performance of the machine learning classifier based on three different datasets; 1) dataset with mixed Arabic dialects, 2) dataset with only gulf dialects, 3) dataset with only emirate dialect. This paper attempt to answer the following research questions;

> ***RQ1*** *which machine learning classifier performs better on dialectal Arabic.*
>
> This paper focuses on dialectal Arabic, and attempt to contribute on expanding the experiments regarding gulf dialects. Two well-known machine learning algorithms are used to answer this question; Support Vector Machine (SVM), and Naïve Bayes (NB). First; we are investigating how these classifiers perform on gulf dialect, and which classifier performs better. Second, we are investigating how these classifiers perform on Emirati dialect, which is one of the gulf group dialects, and which classifier perform better. Moreover, the same classifiers are used to answer other research question.
>
> ***RQ2*** *what is the effect of different preprocessing and different experiment setting on the classification results.*
>
> The preprocessing stage is essential to the sentiment analysis performance. The right preprocessing increases the quality of the data, and thus enhance the classification process. This study is also investigating the impact of two essential preprocessing steps; 1) the removal of stop words, and 2) the use of stemmer. To answer this question, multiple combinations of implementations will be tested.
>
> ***RQ3*** *does the same machine learning model perform better on one Arabic dialects group dataset, in comparison to its performance on dataset with different Arabic dialects groups.*
>
> There is noticeable difference between Arabic dialect groups. For this reason, a tool that is developed for one dialect may not perform well on the other dialect. Therefore, we are investigating the differences of the classifiers performance on one group dialect (gulf), and compare it to the classifier’s performance on a dataset with mix Arabic

dialects. This question is important to learn how important it is to focus the efforts on one dialect group than to treat all Arabic dialects groups as one harmonious data.

RQ4 *does the same machine learning model perform better on one specific dialect dataset, in comparison to its performance on dataset with multiple dialects.*

This research further investigates how important it is to focus on one specific dialect. The question is answered by comparing the performance of the classifier on one specific dialect (Emirati), and compare it to the performance of the classifier on gulf dialect, and again compare it to the performance of the classifier on the mix Arabic dialects. The results will help us understand if considering this level of dialect specific sentiment analysis is important.

1.4 Contribution

This research, in general, contributes in finding the best approaches to perform sentiment analysis on Arabic text data. In particular, this research paper is contributing towards one of the Arabic sentiment analysis challenges, that is the variation of Arabic dialects. This challenge is addressed by many researchers. However, to our best knowledge no research is done on exploring classifier on multi-level dialectal Arabic; mix dialects, one group dialect specific, one dialect specific. The importance of this contribution is the impact of dataset type – in term of Arabic type - on the classifier performance. The outcome is expected to be considered by researchers in future work as in the following;

1) If there is significant difference in the performance accuracy based on different level of dialects, then additional exploration is required to find the best approach that will bring these results close enough, in the multi dialects dataset.
2) If the difference is not significant in the performance accuracy based on different level of dialects, then the major effort in the dialectal Arabic sentiment analysis should be put into enhancing the dialectal Arabic as group, without consideration of one specific dialects.

In other words, we are investigating here, how important it is to focus on dialect specific twitter sentiment analysis.

Furthermore, there are very few researches done on gulf dialects sentiment analysis, even less on Emirati dialect. Among these few researches, most of them are on Saudi dialect sentiment analysis. For this reason, this paper is contributing to exploring sentiment analysis on gulf dialects, as well as on Emirati dialect.

Therefore, the significance of this study could be explained based on two key arguments. First, it explores the machine learning classifiers on different level of Arabic dialects. Second, it contributes to the sentiment analysis on gulf dialects, and on Emirati dialect.

1.5 Thesis Organization

The rest of this research paper is organized as follows. Section 2 discuss Arabic sentiment analysis related work. Section 3 describes the methodology of the study, covering data description, data preparation, data pre-processing, and feature extraction. The experiment and results are presented in section 4. Further discussion is curried on in section 5. Lastly, the study concludes by stating key findings of the research, followed by insights on future work as continuous of this research work.

2 Literature Review

Arabic sentiment analysis is an active research field. Many studies and researches are conducted continuously to contribute to the Arabic sentiment analysis, by either providing new approaches and resources, and/or by enhancing the available approaches and resources. In other word, some researches are done concentrating on building and adjusting the model, some are focusing on data preparation. This section discusses the most prominent studies done recently on Arabic sentiment analysis. In addition a summary of the literature review in available in the Appendix A.

2.1 Lexicon-based Approaches

Some researchers are still exploring the lexicon-based approach. One worth mentioning is the research conducted by Siddiqui, Monem and Shaalan (2018), in which they suggested three ways to enhance Arabic sentiment analysis. The first method is to enrich the lexicon by adding more words from day-to-day communication (informal Arabic words). The second method is avoiding the steps related to data preprocessing. The third step is using the rule-based approach, the heuristics rules in particular, for sentiment analysis. In the experiment, Siddiqui, Monem and Shaalan (2018) used two datasets; twitter data (MSA & Jordanian dialect tweets, labelled into positive and negative equally, 1000 each), and Opinion Corpus for Arabic OCA data (Arabic opinions, labelled into positive and negative equally, 250 each). 360° rules coverage is used as the end-to-end rule chaining principle. These rules consider the position of the polarity in a tweet through the following terms analysis; equal to the text, within the text, ending with the text, and beginning with the text. The performance is evaluated using cross validation and accuracy measurements. The results show 93.9 accuracy on twitter data, and 85.6% accuracy on the OCA data. Siddiqui, Monem and Shaalan (2018) claimed that there is increase by 23.85% in accuracy in comparison with the baseline.

Kaity and Balakrishnan (2019) proposed a framework to automatically generate Arabic sentiment lexicon. Three English lexicons were translated and used in identifying polarity of new words, along with unannotated Arabic corpus. The implementation done on four stages; first preparing seed lexicon, second collecting data and performing pre-processing, third is to find and extract words, fourth is identify the sentiment classification of the words. They claimed that this method is affected and scored better in comparison of other lexicons.

Another interesting experiment is implanted by Areed et al. (2020) where they applied sentiment analysis to client's feedback on online UAE government services (mobile apps). They implement the sentiment analysis on aspect level. The selected aspects are; user interface, functionality and performance, user experience, security and support and update. The dataset size is equal to 2000 after filtering our non-Arabic reviews. The Arabic dataset also highly diverse with MSA and different dialects. The lexicon-based approach was used along with rules-based model. The presented approach is expected to extract aspect, and classify its sentiments. The authors claimed this approach passed the baseline results by 6% in accuracy and 17% in F-measure

On the other hand, Nejjari and Meziane (2019) discussed the limitation of lexicon-based approach. Since this approach depends on the quality of the lexicons, more efforts are required towards building dialectal Arabic lexicons. In some cases, the auto translation of the corpus from English is still used. But this only provide MSA corpus. More words need to be added to the sentiment lexicon to enhance the learning process. The current dictionaries are not optimal. The available lexicon for Arabic language is not comparable to the one available for English.

2.2 Machine Learning Approaches

Most of work conducted using machine learning approaches. In some experiments, machine learning algorisms are used to evaluate the corpus that was built in the experiment. For example, Baly et al. (2019) developed dataset for dialectal Arabic sentiment analysis, and used different machine learning algorithm; including Support Vector Machine (SVM), Logistics Regression, Random Forest Trees, and Ridge Classifier, to evaluate their constructed corpus. The Logistics Regression classifier presented the best results in the study. Similarly, Gamal et al. (2019) applied multiple machine learning classifiers to evaluate twitter dataset for Arabic sentiment analysis after using their proposed methodology in dataset construction. The classifiers used are; Support Vector Machines (SVM), Naive Bayes (NB), Ridge Regression (RR), Maximum Entropy (ME), and Adaptive Boosting (AdaBoost), and the RR performance was the best.

In other studies, the machine learning algorithms is used to evaluate the algorithm performance using different experiment settings; different stemmers, different preprocessing techniques, different feature extraction. For example, in the study conducted by Alomari, ElSherif and Shaalan (2017), where Support Vector Machines (SVM), and Naive Bayes (NB) were explored and compared under different settings of; stemming techniques, N-gram, and TF-IDF features. Another example is the experiment done by DoniaGamal et al. (2018). In their study, they used machine learning for both to evaluate the Egyptian dialect dataset, and to find the best classifier for Arabic dialects. They applied several machine learning algorithms, and the Support Vector Machine (SVM) classifier performed the best with accuracy equal to 93.56%, followed by Logistic Regression with accuracy equal to 93.52%.

El-Alfy and Al-Azani (2020) explored nine different machine learning algorithms on highly imbalanced Arabic tweets dataset expressed in Syria dialect. The dataset is manually annotated, and it is imbalanced with most tweets being negatives. In particular, 75% of the tweets were negative and 25% of the tweets were positive. They used a Word2Vec as feature extraction method. They also evaluated six oversampling techniques. The implementation consisted of two major steps; first they evaluated the classifiers and measured their performances, then they evaluated the classifiers with oversampling techniques and measure their performances to see the impact of oversampling techniques. They used 11 evaluation measurements. The presented results of the SGD (stochastic gradient descent learning) classifier with oversampling yielded the highest score of GM (Geometric Mean) measurement.

In most recent years, deep learning methods is used in Arabic twitter sentiment analysis. Deep learning is more advanced field under machine learning technology. Some studied tried to compare both approaches the traditional machine learning classifier, and the modern deep learning approaches. However, the results are not yet consistence; some researched showed deep learning-based methods are better (Baali & Ghneim 2019), while others showed that machine learning approaches are better (Alayba et al. 2017; Soufan 2019), and in many researches SVM (machine learning classifier), and different deep learning techniques show very close results (Gwad, Ismael & Gültepe 2020; Soufan 2019)

Many researches currently using deep learning techniques are using LSTM techniques. For example, Gwad, Ismael and Gültepe (2020) explored ML and DL techniques; SVM, NB, K-NN, D-Tree and LSTM. LSTM showed best performance (89.8%), followed by SVM (84.70%), and next is the NM (80, 40%). The two others were too low in around 50% accuracy. Authors believe LSTM is more convention technique for Arabic language with its morphology complications. However, we notice the performance of the LSTM is close to the SVM in this research.

Almuqren, Qasem and Cristea (2019) explores different deep learning implementations to compare it with the well-known machine learning classifier; Support Vector Machin (SVM). They selected SVM because it always shows as best classifier in sentiment analysis researches. The results indicate that deep learning methods GRU and LSTM outperformed the SVM classifier. Among deep learning implementations, bidirectional GRU with attention mechanism yielded the best performance.

On the other hand, Soufan (2019) explores different classic Machine learning algorithms which are Support Vector Machine (SVM) and multinomial Naive Bayes (MNB). They also experiment with Deep Learning methods which are Long Short-Term Memory (LSTM), Word-Level CNN, and Character-Level CNN. The presented results were very close when comparing the performance of machine learning algorithms and the performance of the deep learning methods. Nevertheless, the machine learning classifiers showed a slightly better accuracy than the results of deep learning methods. Similarly, Alayba et al. (2017) performed sentiment analysis experiment on domain specific tweets (health care). They, first, collected data using twitter API and health care key words from trending hashtags such as “closing hospital”, ”improving health”, and “your opinion about health”. Then, the tweets were annotated into positive, negative and neutral. The experiment was done using different ML algorithms and deep learning method. Th best performing classifier was SVM.

Machine learning is shown better results in comparison to other approaches. However, Nejjari and Meziane (2019) pointed out some limitations. Machine learning perform better when used on domain specific data. This might be because other approaches depend on the quality of the sentiment lexicon. One is the efforts required in labelling the tweets; the more labelled data is available, the better the classifier performance is. Additionally, handling the negation words is still considered an open problem

2.3 Hybrid-based Approaches

Furthermore, Arabic sentiment analysis researches also conducted using hybrid-based approaches in attempt to enhance the classification performance. An experiment conducted by Al-Twairesh et al. (2018) sentiment analysis of Saudi dialect tweets used hybrid approach; corpus-based and lexicon-based approaches. In this experiment they focused on feature selection methods. Different classifiers were used which are; two-way classifier, three-way classifier and four-way classifier. The two-way classifier showed best F1-score result.

Another study conducted using the hybrid approach is done by Al-Harbi (2019). Author used sentiment lexicon that consisted of 3400 sentiment terms, 580 compound phrases, and popular English original words written in Arabic such as نايس، لايك. Besides, four machine learning algorithms were used; SVM, NB, Random Forest, K-NN". For feature selection, they used methods such as "Correlation-based Feature Selection, Principal Components Analysis, and SVM Feature Evaluation". Nine features were explored such as positive words number (PWN), negative word number (NWN), and Negation words number (NgWN). SVM classifier attained the highest accuracy = 92.3%.

The discussion on the hybrid approaches by Nejjari and Meziane (2019) stated that the results of hybrid approaches are promising. While the problem that most researchers faced is the unavailability of dialectal lexicons.

2.4Arabic Sentiment Analysis Enhancement Experimentations

The quality of the dataset is very important for the classifier training and classification performance. A research is done by Gamal et al. (2019) proposed the methodology for constructing the twitter dataset for Arabic sentiment analysis. The extracted dataset consists of labelled tweets expressed in MSA and in Egyptian dialect. The proposed process included 12 steps; 1) collect tweets, 2) remove non-Arabic characters, 3) tokenize, 4) remove stop words, 5) remove repeated letters 6) remove URLs and users mention, 7) remove hashtags and retweets, 8) remove diacritics, 9) handle emoticon, 10) normalize letters, 11) label tweets, 12) adjust data skewness. To evaluate the proposed method, Gamal et al. (2019) used five different machine learning classifiers, with TF-IDF feature extraction method. The performance of the model was evaluated using cross validation. The ridge regression classifier achieved the highest accuracy score of 99.90%.

Furthermore Hegazi et al. (2021), suggested an integrated approach as pre-processing solution for the Arabic sentiment analysis. This approach considers four stages; data collection, text cleaning and normalizing, text enriching, and result presentation. They recommended connecting to social media source for data streaming through APIs. The received data should undergo text cleaning and text normalization. Additional step of text enrichment includes; tokenization, stemming root, generation, and morphological generation. After that, the data is expected to be ready for analysis and information extraction. In this study, authors experimented their proposed method on twitter data.

Study conducted by Goel and Thareja (2018) suggested the use of hashtags to extract emotions, as an enhancement step to the sentiment analysis. They divided emotions into four categories; Happy-active, Happy-inactive, Unhappy-active, Unhappy-inactive. Nine keywords were identified as follow; "happy" and "excited" for the Happy-active category, "relaxed" and "sleepy" for the Happy-inactive category, "angry" and "afraid" for the Unhappy-active category, and finally "tired", "bored", and "sad" for the Unhappy-inactive category. These keywords were used to extract 1000 tweets for each keyword, when they are used as hashtags. For example, in this tweet "feeling left out… #bored", the hashtag is enough to represent the emotion of the tweet. However, this is not the case with all tweets. For example, in the tweet "first-ever Angry Birds World to open at DFC #angry#birds", the keyword angry is a hashtag that doesn't indicate the author of the tweet is angry. For this reason, they used emotion lexicons as well to enhance the emotion analysis performance. Goel and Thareja (2018) also investigated the effectiveness of using the identified keywords. They extracted general tweets that has hashtags. Then they counted the frequency of each keyword hashtag to find the distribution of the emotions over 2000, 5000, 10000, and 20000 tweets. The results show that the keyword "sad" is the most expressed while the keyword "afraid" is the least expressed.

Baali and Ghneim (2019) studied emotion analysis of twitter data, rather than the polarity classification. They claimed that emotion analysis is a deeper analysis. Four key emotions were identified; happiness, anger, fear and sadness. The dataset consisted of 5600 pre-labelled tweets, 1400 tweets for each key emotion identified. For each key emotion data, the dataset was split into 90% training and 10% testing. The experiment is done using deep learning approach; the Convolutional Neural Networks (CNN) with the word vectors training technique. The word vector was trained, first, to create word2vec model. Then CNN classifier was trained through four steps; word vectorization, sentence vectorization, document vectorization, and then classification of emotions (anger, joy, sad, fear). Twitter emotion analysis is also explored by Manguri, Ramadhan and Amin (2020) on the current event of COVID-19. The data are gathered in seven days from twitter using two hashtags; #COVIS-19 and #coronavirus. The total data size is 530232 tweets. To perform the emotion analysis, ten keywords were selected. And to measure the polarity of these emotions, they developed "Emotional Guidance Scale", where each of the emotion is assigned a score from 1 to -1. The highest emotion for example is happy and joy assigned to the highest score that is 1. The lowest emotion is depressed so it gets -1. The overall results show that around 60% people feels calm. In addition, 36% are positive toward the event, while 14% are negative, and around 50% feeling neutral.

Hammad and Al-awadi (2016) focused on finding the best approach for Arabis sentiment analysis in term of "lightweight". The collected data consists of 2000 social media Arabic reviews. These reviews are domain specific related to customers reviews on Jordanian hotels, and collected from Facebook, Twitter, and YouTube. They used four classifiers; Support vector Machine (SVM), Naïve Bayes, Decision Tree and Back-Propagation Neural Networks (BPNN). The highest accuracy achieved by SVM (96.06%). To evaluate more, the experiment was conducted on different size of dataset; 300, 600, 900, 1200, and 1500. Interestingly, the results of F-measure showed that SVM consistently learning better with more training data. While BPNN is not learning well as F-measure score is getting low with more data. Authors explained this result because the dataset is ambiguous. Furthermore, training time was evaluated, the results showed SVM train data is shortest time (6.45seconds), followed by Naïve Bayes (11.41 seconds)

Besides, some research attempted to study and evaluate the different features selection and processing techniques to find the best settings for sentiment analysis applications. Oussous, Lahcen and Belfkih (2019) explored the impact of using different stemming techniques. They conducted experiment with three machine learning classifiers where unigram is used as feature. Several experiments conducted with no stemming, and with different stemmers applications; Motaz, Light stemmer, ISRI, Khoja and Tashaphyne. The best results attained by Support Vector Machine (SVM) with no stemming. In their study they suggest that stemming doesn't improve the performance of the classifiers in case of dialectal Arabic is used. Additionally, when comparing different stemmers and stemming approaches; the light stemming performed better than root extraction techniques with all classifiers.

Baly et al. (2019) suggested enhancing the sentiment analysis is to use multi-way sentiment analysis. This means there are more classes than the regular three; positive, negative, and neutral classes. In one study, 5-scale points is used, where the following classes are used; very positive, positive, very negative, negative, and neutral. This step required additional efforts on labelling dataset (Baly et al. 2019)

The research done by Farha and Magdy (2019) is worth mentioning as example of releasing Arabic sentiment analysis resources free for the public. Farha and Magdy (2019) developed an online open-source tool for Arabic sentiment analysis, they called it "Mazajak". The tool is based on deep learning approach along with word embeddings technique. Word2vec was used in this implementation for word embeddings, and the embedding size (D) is set to 300. Authors claimed this is the largest word embeddings for Arabic as it uses 250M unique tweets. The used corpus aimed to cover wide variety of topics and dialects; therefore, the data collection duration was almost three years (2013 – 2016). For the model Architecture, both CNN and LSTM are used, with multiple layers including; max pooling, dense layer, and softmax. They tested their proposed model on different datasets. They said their system attained state-of-the art results with different Arabic dialects data. They published their tool online (Mazajak.inf.ed.ac.uk:8000) for research purposes. The tool provides three level of sentiment analysis; simple text input, file level, and time line sentiment analysis. In addition, online API module is provided to support other researches.

Almuqren, Qasem and Cristea (2019) presented an interesting experiment in applying sentiment analysis on customer satisfaction for different telecommunication companies in the Saudi Arabia; STC, Mobily, and Zain. The total dataset consisted of 20,000 tweets. To collect relevant tweets, they used hashtags such as #Zain #mobily #STC. They filtered tweets by location to select Saudi tweets. In this research, Support Vector Machine (SVM) machine learning classifier was used, as well as "two deep learning approaches: long short-term memory (LSTM) and gated recurrent unit (GRU)." They implemented LSTM and GRU in two ways; in the first implementation, they added the attention mechanism, in the second implementation, they performed character encoding. The best performance was by "bidirectional-GRU with attention mechanism". The best classifier then used to predict customer satisfaction for each telecommunication company, the results of predicted satisfaction were very close to the actual satisfaction.

Al-Saqqa , Obeid and Awajan (2018) used ensemble learning technique on small dataset to enhance the sentiment analysis performance. In the ensemble learning multiple classifiers are used together to perform sentiment analysis task. Authors used the following classifiers; Naïve Bayes (NB), Support Vector Machine (SVM), k-nearest neighbors (K-NN), and Decision Tree (DT). In the first experiment, they run these classifier alone on the dataset to measure the baseline performance. Then, the ensemble learning is performed as different classifiers combination is explored, with diffirent use of unigram and bigram features. The results presented two good combination. The first one is the use of NB, SVM, and K-NN classifiers with unigram feature selection. The second is the use NB, SVM, and DT classifiers with bigram feature selection. In comparison with baseline performances, the ensemble technique showed better results than all other classifiers except SVM.

2.5 Dialectal Arabic sentiment analysis

Some researches explored sentiment analysis on dialectal Arabic. We observed that most of work contributed to the Egyptian dialect and Levantine dialects in comparison to other dialects. The study (Gamal et al. 2019) is conducted on MSA and Egyptian dialect. While (Baly et al. 2019) experiment sentiment analysis in regards to the Levantine dialects. On gulf dialect group, most researches are noticed done on Saudi dialect (Al-Twairesh et al. 2018). This probably because the most active Arab internet users are from; first Saudi Arabi, then from Egypt, followed by Algeria, then the United Arab Emirates (Alruily 2020).

Some studies done in contribution to provide and to enhance of the dialectal Arabic sentiment analysis resources. One example is the research conducted by Baly et al. (2019). They provided Levantine dialect corpus (ArSenTD-LEV), consisting of 4000 tweets. The annotation of the data is done in consideration of the topic, the target of the sentiment, and the sentiment of the tweet. Different ML algorithms were used to evaluate the corpus in which Logistic regression showed better results. Authors claimed that the enhancement of the annotations enhanced the classifier performance by 10% in comparison with the baseline. However, the topics were analysed manually on small sample of 200 tweets, which might not be relevant on big scale of data.

Mdhaffar et al. 2017 presented the Tunisian sentiment Analysis Corpus. The corpus size is 17,000, collected from Facebook comments expressed in Tunisian dialect and annotated into positive and negative classes. To evaluate the presented corpus, they applied machine learning classifier; Multi-Layer Perceptron (MLP), Naive Bayes classifier, and SVM. on the available corpus; MSA and other Arabic dialects. Then, they used similar classifier on the TSAC. The author claimed that the results of TSAC was better than the other datasets. Similarly, another study explored sentiment analysis on Tunisian dialect is conducted by Jerbi, Achour, and Souissi (2019).

Alomari, ElSherif and Shaalan (2017) presented the Jordanian General Tweets (AJTC) corpus. The data is labelled positive and negative. Two machine learning algorithms were used Support Vector Machine (SVM) and Naïve Bayes (NB). They explored different use of stemming and N-grams. They concluded that the best results were performed by SVM with stemming, using bigrams, and TF-IDF features, with accuracy equal to 88.72%. They also claimed that it outperformed other related work. Similarly, Al-Harbi (2019), and Atoum and Nouman (2019) explored sentiment analysis on the Jordanian dialect.

Alruily (2020) prepared and provided the dialectal Saudi Twitter corpus. The corpus size is 207452 tweets expressed in Saudi dialect. Additionally, they performed interesting analysis in which they compared the corpus of Saudi dialect, with Egyptian corpus as well as the MSA corpus. They highlighted some differences on challenges level which emphasizes on looking at each dialect as a special case, as each dialect has its own different characters and required particular attention.

Al-Thubaity, Alqahtani and Aljandal (2018) introduced Saudi dialect lexicon (SauDiSenti), for twitter sentiment analysis. The lexicon consists of 4431 words and compound phrases; 24% positive and 76% negative. They also introduced a dataset consists of 1500 labelled tweets. To evaluate the SauDiSenti lexicon, they calculated the precision, recall and F- measure. Furthermore, they compared it with Arabic sentiment lexicon called AraSenTi. They conducted two experiments; first with only two sentiment classes (positive and negative), second with three sentiment classes (positive, negative, neutral). The results show the AraSenTi performed better in the first experiment, while the SauDiSenti performed better in the second experiment.

One feasibility study focused on Emirati dialect sentiment analysis (Al Suwaidi, Soomro & Shaalan 2016). In this study, authors explored the possibility for a word from Emirati dialect to be assigned with a sentiment score. The results were promising, and further investigation is required through full sentiment analysis experimentation.

To summarize, while surveying some recent researches conducted on Arabic sentiment analysis, there are still many challenges exist which required more efforts. One of the most discussed and faced issue with Arabic sentiment analysis is the Dialectal Arabic. Technically, machine learning is the most used approach in sentiment analysis, we also noticed that SVM and NB classifier are the most used classifiers in the machine learning approaches which show best results in most cases, SVM in particular.

3 Dataset and Methodology

This section demonstrates the methodology followed in our experiment. In particular, it describes the datasets used in the experiment, dataset preparation, data preprocessing, and feature extraction method. We used supervised machine learning approach where labelled data is required, and machine learning models are required to perform the classification model.

Figure 1 presents the framework we are following in this experiment. First step is datasets preparation where three of the required datasets were prepared; the mix dialects datasets, the gulf dialect datasets, and the Emirati dialect datasets. The next step is the preprocessing steps; data normalization, tokenization, light stemming, and Stopwords removal. After that, the feature extraction steps, and then the major step of machine learning classification where data is trained to build the classification model, and test data is used to test the data performance. Finally, the machine learning model is evaluated. This framework – that we are using - is the basic framework for supervised machine learning sentiment analysis. This study is trying to evaluate the performance of classification on different datasets. Therefore, we are following the standard stages of the supervised machine learning approach.

3.1 Dataset Description

The dataset we are using in this paper is twitter data. In this experiment, we are working with different Arabic dialects levels; from general (Arabic dialects), to group dialect specific (Gulf), to one dialect specific (Emirati). For this reason, we used Twitter Arabic Dialect Dataset (TAD). TAD is previously collected twitter data that consists of over one million tweets. The tweets are expressed in different Arabic dialects; Egypt, Gulf, Levant, North Africa. Each tweet is labelled by the dialect group it belongs to, as well as its polarity; positive, neutral and negative. We used TAD file to generate our three datasets required from this experiment.

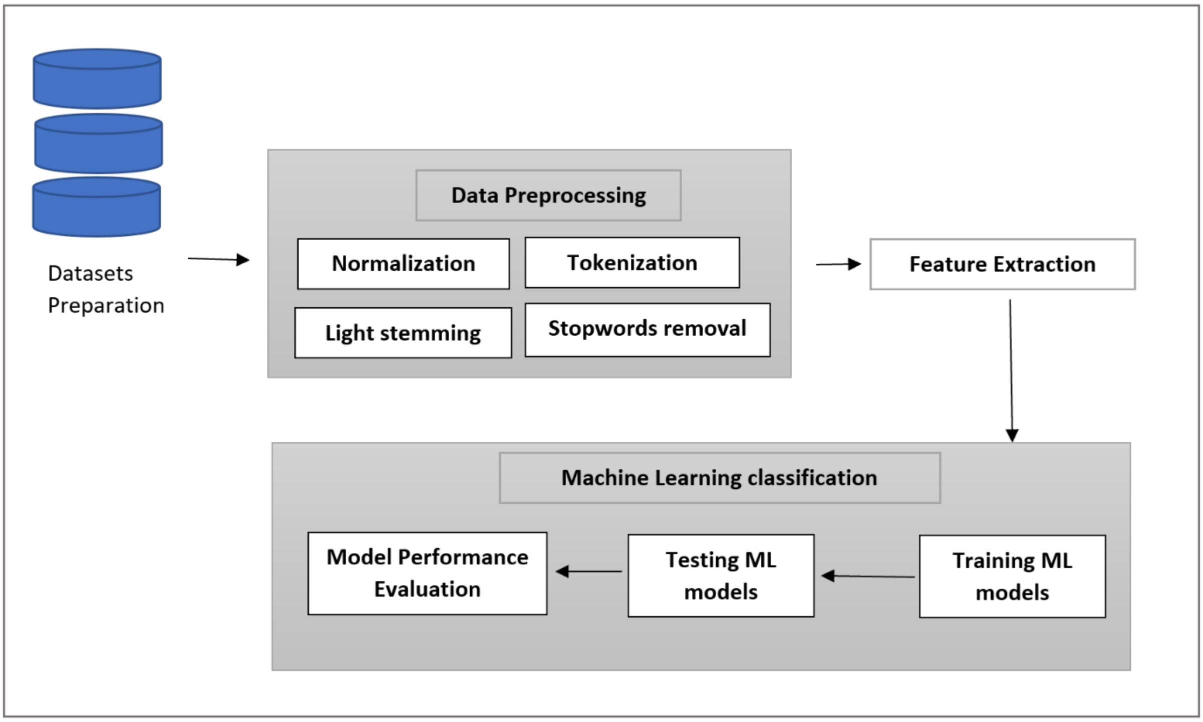


***Figure 1**: Sentiment Analysis framework*

3.2 Data Preparation

In this experiment, three datasets required. The first dataset is the mix dialect tweets dataset which has mix Arabic dialects; Egypt, Gulf, Levant, North Africa. The second dataset is the gulf dialect tweets dataset. The last dataset is the Emirati dialect tweets dataset. In this section, a description of each of the required dataset is explained.

3.2.1 Mix dialects dataset

The first dataset required in the experiment is the mix dialects dataset. This dataset contains different Arabic dialects. We could use the original file for this experiment, however the tools and memory we are using can't handle this large data file size. This also might cause a dramatic difference between the three datasets in size.

To prepare this dataset, we could randomly extract data from the TAD file. However, we wanted to ensure that the dataset contain tweets from all different dialect group. Therefore, we extracted 2500 tweets randomly from each dialect groups. So, our total dataset size = 10,000 tweets. However, the size changed after data processing and removing neutral tweets, as shown in table 8. Some examples of the tweets in this dataset are shown in table 3

Mix dialects tweets
ترا بموت حماس ايش يصبرني لين ٢٧ 😭😭😭💔💔💔
توني ادري ان باجر عندي امتحان حتى مادري شنو الصفحات 🙂
هو انا مش هدخل الكورس انهارده كمان ولا اي 🤭🙄
صور بزاف توجع 💔 من الحروب والهجرات
وايد كيوت 😫
هو انا ليه متخلقتش كوريه 😣😭

__Table 3__: examples of tweets from mix Arabic dialects datasets

3.2.2 Gulf dialects dataset

Tweet examples from Gulf dialects
ليش وايد كيوتت😭😭
اشهالأخبار 🙄 يعني شنهو استفاد القارئ من خبر مثل چذي 🤭
صدق الوقت مره يمشي بسرعه شدعوه اصبر اخلص اشغالي لا تخلص ياخي 😭😭
اموت فيج واجد 💗
ماحد زعلني ولا احد كفو يزعلني 😂😂😂😂
بداوم بدون غتره وعقال و مستحي 😅

__Table 4__: examples of tweets from gulf dialect group datasets

The second dataset required in the experiment is the gulf dialects dataset. Gulf dialects is spoken by six countries; UAE, Saudi Arabia, Kuwait, Bahrain, Qatar, Oman and Yemen. Each of these countries has its own dialect; the Emirati dialect, the Saudi dialects .... etc. The dialects are similar, however every country has its unique pronunciation of some words, or has its own words. to create gulf dialects datasets, we randomly extracted 10,000 tweets from the “gulf” dialect filter, and saved it as csv file. Table 4 presents some examples of tweets from the gulf dialect group datasets.

3.2.3 Emirati dialects dataset

The Third dataset that we need for this experiment is the Emirati dialect dataset. The reason part of the experiment is important because there are some major differences in the dialects between different golf regions. For example to say what do you want in different gulf dialects is as follow; in Saudi dialect (ايش تبغي), in Emirati dialect (شو تبا), and in Kuwaiti dialect(شنو تبي).

It was not straightforward to extract the tweets related to Emirati dialect. To do so, we were inspired by the previous research conducted by Al Suwaidi, Soomro and Shaalan (2016) in sense of the selected Emirati phrases. In fact, we initially wanted to use the same phrases identified in their research. However, after analysing the extracted tweets, we find out the following

1) The word (جزاك) is only Emirati when considering the context, and how it is pronounced. However, if used to extract the tweets, many irrelevant tweets are expected to show up because the word is widely used in Arabic dialects in general. For example in the popular phrase "جزاك الله خير"
2) some of the selected words are used in Emirati dialect but also in other gulf dialects. For example; (اسولف), (عبالكم), (تحقر), (بهالعمر), (تحقر), (ماتستحي), and (وهق)
3) The other three words; (اندوكم), (متفيج), and (يغربل) are mainly Emirati phrase and will serve us in selecting the Emirati tweets. So, we are using these three Emirati phrases as our keywords to generate the Emirati dialect datasets.

In addition, we added seven more words and phrases from Emirati dialects that we believe are mostly used by Emirati Internet users than other gulf internet users. In total we used 10 Emirati phrases to find the relevant tweets. Table 5 below presents the complete list of the selected Emirati phrases, with their English meanings and pronunciation. It also shows the retrieved tweets against each phrase from total of 235,451 gulf dialect tweets.

#	Phrase	English Synonym	Pronunciation	Total	Total (cleaned)
1	يرمس	talk	*yarmes*	306	293
2	غترة	men traditional scarf	*gatra*	172	141
3	يزقر	to call someone	*yezger*	25	10
4	اندوك	here you have it	*endok*	28	26
5	متفيج	has time	*metfayej*	71	67
6	يغربل	Disorganized/ Garble	*yegarble*	23	21

7	زاهب	ready	*zaheb*	8	8
8	أروم	I can	*aroom*	526	277
9	برايه	let it be	*brayah*	11	11
10	خلاف	then	*khelaf*	20	6

***Table 5**: list of selected Emirati words and phrases*

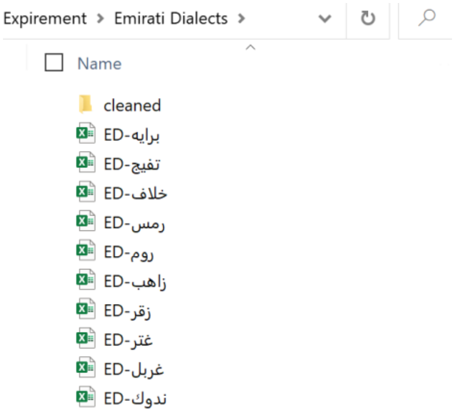
There are many variations of the selected words and phrases. To select the various derivations of the word, we searched by only the basic letters of the word or phrase. So that we can get all the related variations. Table 6 shows the exact phrase used to extract the tweets.

#	Emirati Phrase	search term	derivations examples
1	يرمس	رمس	يرمسون، ترمس، الرمسه، رمسني
2	غترة	غتر	غتره ، غترة، غترته
3	يزقر	زقر	زقرني، تزقره، يزقرها، بنزقرلك
4	اندوك	ندوك	اندوك، اندوكم
5	متفيج	تفيج	متيفجة، متفيجين، بتفيج
6	يغربل	غربل	يغربله، غربلها، غربلني
7	زاهب	زاهب	زاهبه، زاهبين،
8	أروم	روم	اروم، تروم، ماروم، يرومون
9	برايه	برايه	برايها، برايهم،
10	خلاف	خلاف	خلاف

***Table 6**: list of terms of Emirati words and phrases used to filter the tweets*

For filtration we used “text filters” feature in Excel, then moved the resulted records to another file. The total extracted tweets using the selected ten phrases from Emirati dialect are 1190. The extraction processes done separately for each phrase, and the result of each phrase is saved in separate file. Additional step is required that is to ensure the quality of the extracted tweets.

The Emirati dialect tweets extraction process done separately for each phrase, and the result of each phrase is saved in separate file as seen in figure 2. This enabled us to easy perform data validation on each data tweets set. This is important because mostly each dataset of one phrase/word has similar problems such in the case the phrase (روم) which is also part of the popular word (رومانسي) and the word (روما). We had to remove almost 50% of the retrieved



cleaned
ED-برايه
ED-تفيج
ED-خلاف
ED-رمس
ED-روم
ED-زاهب
ED-زقر
ED-غتر
ED-غربل
ED-ندوك

***Figure 2**: Extracted Emirati dialect tweets*

tweets. That is why the total reduced dramatically from 526 to 277, in comparison to other phrases. The word (خلاف) is also used in general Arabic dialects and also in MSA for a different meaning which is “conflict”, while in Emirati dialect it is used to mean “later” or “after that”. However, with other phrases, the phrases brought good results in representing the Emirati dialect. we went through each file and validated the quality of the tweets. In general, the following corrections, and validations were made;

1) Removed non-Emirati dialect tweets.
2) Removed tweets that have similarly written words but different meaning
   One example is the word (ترمس) which also means “coffee pot”, in Saudi dialect, while in Emirati it means “she talks/ she is talking”
3) Corrected miss-spelled words.
4) Labelled the tweets.
   Because we wanted to ensure that each tweet is labelled correctly, we labelled them manually by three Emirati dialect native speaker annotators, and because the dataset is relatively small, the annotation process is done through discussion among the annotators in one session.
5) The last step is constructing the Emirati dialect is to integrate all files into one data file, and saved it in csv dataset file.

Table 7 present some examples of the tweets from Emirati dialect dataset.

Tweet examples from Emirati dialects
مو متفيج للجم احس اليوم بس بعد بسير 🙂
الواحد ما يقدر يرمس ولا يقول شي 👎
انا اروم اغيب بس المشكله امتحان فيزياء اليوم 💔
اخس شي الجو حلو وانا فالدوام ما اروم اشرد 💔
عندي واحد بس مو مالي لو هاه كنت اروم اعطيج 😭
بداوم بدون غتره وعقال و مستحي 😅

Table 7: examples of tweets from Emirati dialect datasets

After selecting and cleaning the datasets, we decided to use two classes; positive and negative. Therefore, tweets with the neutral class were removed from all datasets. It affected the original dataset size we intended to use. The final dataset size in shown in the table 8.

Dataset	size	Positive tweets	Negative tweets
Mix dialects Tweets	13982	6939	7043

Gulf Dialect Tweets	14000	8000 (57%)	6000 (42%)
Emirati Dialect Tweets	500	239 (48%)	261 (52)%

Table 8: *datasets size*

3.3 Data Preprocessing

Data preprocessing is an essential stage in sentiment analysis tasks, because the quality of data is crucial to the performance of the sentiment analysis classification. Poor quality data will lead to poor results. Data preprocessing attempts to eliminate noice and inconsistencies in the text data. In this experiment, we followed the standard data preprocessing steps as shown earlier in figure 1. The preprocessing stage – in our experiment - included; normalizations, data cleaning, stop words removal, tokenization, stemming and feature extractions.

Normalization is an essential step in data preparation, especially when dealing with unstructured data. It changes the text data into more standard format. Letters Normalization targets some letters that could be used in different forms either because of the variations of the letter or because of misspelling. For example, the letter (أ) has many variations such as (ا), (إ), (آ). All changed in this case to the simplest form (ا). Another example of letter normalization for Arabic text data is to remove the diacritics. In fact, all noise in the text data were previously removed, these noises include; user mentions (@), hashtags (#), URLs, retweets, punctuations, symbols and characters such as $%?! …

Tokenization is the process of splitting given sentence in to smaller segments (tokens). In our experiment, the tokenization split each tweets text into words using spaces between words as marker of start or end of a word. The uniqueness of the words is also ensured in the process. We used the tokenizer module from NLTK (Natural Language Tool Kit) library in python:

```
import nltk

from nltk.tokenize import word_tokenize
```

Stemming is eliminating features process where each word is shorten to its stem (root). Generally, the stemming can be implemented by two ways. The first is the light stemming where the prefixes and suffixes of the words are deleted. The second is the root stemming where the main goal is to extract the root of the words.

The stemming step is one of the preprocessing step to be investigated in this study. Therefore, the use of stemming is going to be performed on half of the experiment, while the other half there will be no stemming. As later we will analyse and study the impact of using stemming in the preprocessing stage. And because of the complexity of the Arabic language, we selected light stemmer as a safe option. For stemming, we used the stemmer from the tool called tashaphyne that provide Arabic Light stemmer. Tashaphyne is a python library that performs light stemming on Arabic text in that it removes the suffixes and prefixes of the words (Zerrouki 2019). To use the tashaphyne stemmer, we first installed the library in python using PyCharm, then we import it, and then made the following function "stemming(s_text)", to recall it when required;

```
from tashaphyne.stemming import ArabicLightStemmer
ArListem = ArabicLightStemmer()


def stemming(stem_text):
   n_text=[ArListem.light_stem(word) for word in stem_text]
   return n_text


dataset['text_final']=dataset['text1'].apply(lambda x: stemming(x))
```

As for emoticons, we decided to keep emoticons as they give additional sentimental meaning to the tweets, and in some cases, we classify the tweet only based on the emoji added next to the words, such as in the following example;

#	Tweets example	Classification
1	اندوكم هذا ☺	Negative
2	اندوك 🌹🌻	Positive
3	غتره ☹ ♡	positive

Table 9*: emotions tweets example*

Removing stop words is another popular and important task in sentiment analysis preprocessing phase. Stop words removal is the processes of eliminating widely used words in the sentences that doesn't add a particular meaning or value such as the use of conjunctions or pronouns. In our case, we didn't use the Arabic stop words function from the popular python library NLTK, because its more suitable for MSA and doesn't work well with dialect Arabic. we searched on the Internet for available stop words list, however the contents are not very relevant. Therefore, we looked at random tweets in our datasets and added many words that thought of as stop words along with the one we found on the Internet. There are over 500 words in the stop words list we used. Some examples are ('ترا'), ('أيضا'), ('بس'), ('الى'), ('وهذا'). Full list in shown in Appendix B in this research paper. To use this external list, we first loaded the file in python working environment after converting it to csv file, then we stored its content in a variable, through a function, so we can use it anywhere needed to remove the stop words.

```
my_stop_words = pd.read_csv(r"sw_list.csv", encoding='utf-8-sig')
with open('lsw_ist.csv', 'r', encoding='utf-8') as sw:
        sw_content = sw.read()
        mystopwords = set(sw_content.split())
```

3.4 Feature selection

There are many methods for feature extraction. In our experiment, we applied the TF-IDF feature extraction method. TF-IDF is short for "Term Frequency-Inverse Document Frequency. It mainly calculates the relevance of the word to the document. In our case, it calculated each token's frequency in every tweet. Therefore, it learns how important each token is to the tweets. TF-IDF is widely used in the sentiment analysis tasks, as seen in section 2 of this paper. We applied TF-IDF Vectorizer through sklearn python library:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

4 Experiment and Results

The experiment is conducted in this research paper to answer the stated research questions in section 1.3 of this study. Now that we prepare the data, the datasets and the setting are ready to run the experiment. In this section we will discuss; tools and libraries used to perform the experiment, the selected machine learning algorithms, the classification process, and performance evaluation along with results presentation. We are using supervised machine learning approach as seen in figure 1.

There are total of 12 experiments performed for each classifiers. Two classifiers are used which as SVM and NB. There are three different datasets to be used which are mix dialects dataset, gulf dialect dataset, and Emirati dialect dataset. Moreover, for each dataset and for each classifier, different experiment settings were adjusted; with stemming, without stemming, with stop words removal, without Stopwords removal. Therefore, there are eight experiments per datasets. For each dataset, two models are tested with the following settings;

1. With stemmer, with Stopwords removal.
2. With stemmers, without Stopwords removal.
3. Without stemmer, with Stopwords removal.
4. Without stemmers, without Stopwords removal.

4.1 Tools and libraries

Python is one of the most used programming languages for sentiment analysis and Natural language processing tasks (Zahidi, Younoussi, & Al-Amrani 2021). Therefore, to run our experiment, we used python 3.9 programming language, along with PyCharm as our IDE (integrated development environment) to be able to work easilty with python. PyCharm makes it easy to install library and run python code. Also, its very practical to deal with files in PyCharm. We relied on python libraries for most of the required tasks in the process. Some python libraries are used for data manipulation such as; pandas, numpy and string. NLTK python library is used for text processing and methods. While sklean is used for the machine learning tasks. We also used excel in constructing the Emirati dataset, mainly in finding and filtering tweets.

4.2 Machine learning classifiers

For this experiment we selected two popular machine learning classifiers for the sentiment analysis classification which are; the Support Vector Machine (SVM) and Naïve Bayes (MNB).

As shown in the literature presented in section 2, these machine learning classifiers were selected because they attained best results in the most experiments done using machine learning approaches. As well as, they are widely used for Arabic sentiment analysis.

4.3 Training and Testing Classifiers

In the classification process, classifiers are used to define the predefined classes of the input data. For this task, two supervised machine learning algorithms were used to classify data; SVM and NB. Only two classes are used in this work; positive and negative. For the classification process, the dataset is split into 80% training dataset and 20% testing dataset where the classifier uses the training data to learn from the labelled data, and use the other dataset, the testing dataset, to test and evaluate the performance of the classifier. The table 10 present – for each dataset - the number of tweets that are used for training, and the number of the tweets used for testing using the 80/20 split method. We actually, attempted to use different splitting ration such as 70/30 and 90/10, but we noticed it has no significance effect on the classification process.

Dataset	Training data	Testing data
Mix Dialect Tweets	11,186	2,796
Gulf Dialect Tweets	11,200	2,800
Emirati Dialect Tweets	400	100

***Table 10**: number of tweets used as training data, and test data*

To implement this in python, we used the "train_test_split" function from sklearn library;

```
Train_X, Test_X, Train_Y, Test_Y = model_selection.train_test_split(dataset['text2'],
dataset['sentiment'], test_size=0.2)
```

We also used TF-IDF feature extraction in our experiment. We used TF-IDF vectorizer to vectorized the words in our dataset. This purpose is to measure the importance of a single word in the data file, as compared to the whole data file (document). We implemented this process using python's library sklearn as follow

```
Tfidf_vect = TfidfVectorizer(max_features=100000)

Tfidf_vect.fit(dataset['text_final'])
```

We set the max features to 100,000 after we experimented with different values. We noticed that increasing of the max features enhanced the performance of the classifiers. However, that is true only when we increase it from 5000 to 10,000, to 100,000. While the value 500,000 of the max features started to lower the performance.

4.4 Evaluation and Results

There are many various methods to evaluate the performance of the classification models. Accuracy is one of the widely used method for evaluation, which we used to evaluate our experiment. The accuracy calculates the number of correctly classified tweets over the number of all tweets, as in figure3. Table 11 demonstrate the elements of the accuracy formula.

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

***Figure 3**: Accuracy measurement formula*

Term	Meaning	Description
TP	True Positive	all tweets that are correctly classified as positive
TN	True Negative	all tweets that are correctly classified as negative
FP	False Positive	all tweets that are incorrectly classified as positive
FN	False Negative	all tweets that are incorrectly classified as negative

***Table 11**: Accuracy evaluation demonstration*

The accuracy results of classification models of twelve experiments are presented in table 12, and a visual representation of the results is shown in figure 4. During the experiment we run the models multiple times, to ensure that we got the best score from the classifier under the tested settings and dataset. The best performance is shown with gulf dialect dataset, when no stemming is used and without removing stop words. This best performance is attained by the SVM classifier. The models performed better when gulf dialect dataset is used. While the model’s worst performance was with Emirati dialect dataset.

Dataset	Preprocessing		Accuracy	
	Stemming	stop words	MNB	SVM
Mix Dialects Tweets Size = 13982	with Light Stemming	with Stopwords removal	72.22%	69.93%
		without Stopwords removal	70.65%	70.61%
	No Stemming	with Stopwords removal	72.04%	71.97%
		without Stopwords removal	**72.99%**	72.33%
Gulf Dialect Tweets *Size = 14000*	with Light Stemming	with Stopwords removal	71.18%	72.89%
		without Stopwords removal	72.50%	73.32%
	No Stemming	with Stopwords removal	74.57%	75.14%
		without Stopwords removal	74.86%	**75.36%**
Emirati Dialect Tweets *Size = 500*	with Light Stemming	with Stopwords removal	66%	68%
		without Stopwords removal	62.00%	64.00%
	No Stemming	with Stopwords removal	69.00%	**70%**
		without Stopwords removal	64.00%	67.00%

***Table 12**: Accuracy of ML classifiers for each dataset, cross-different settings*

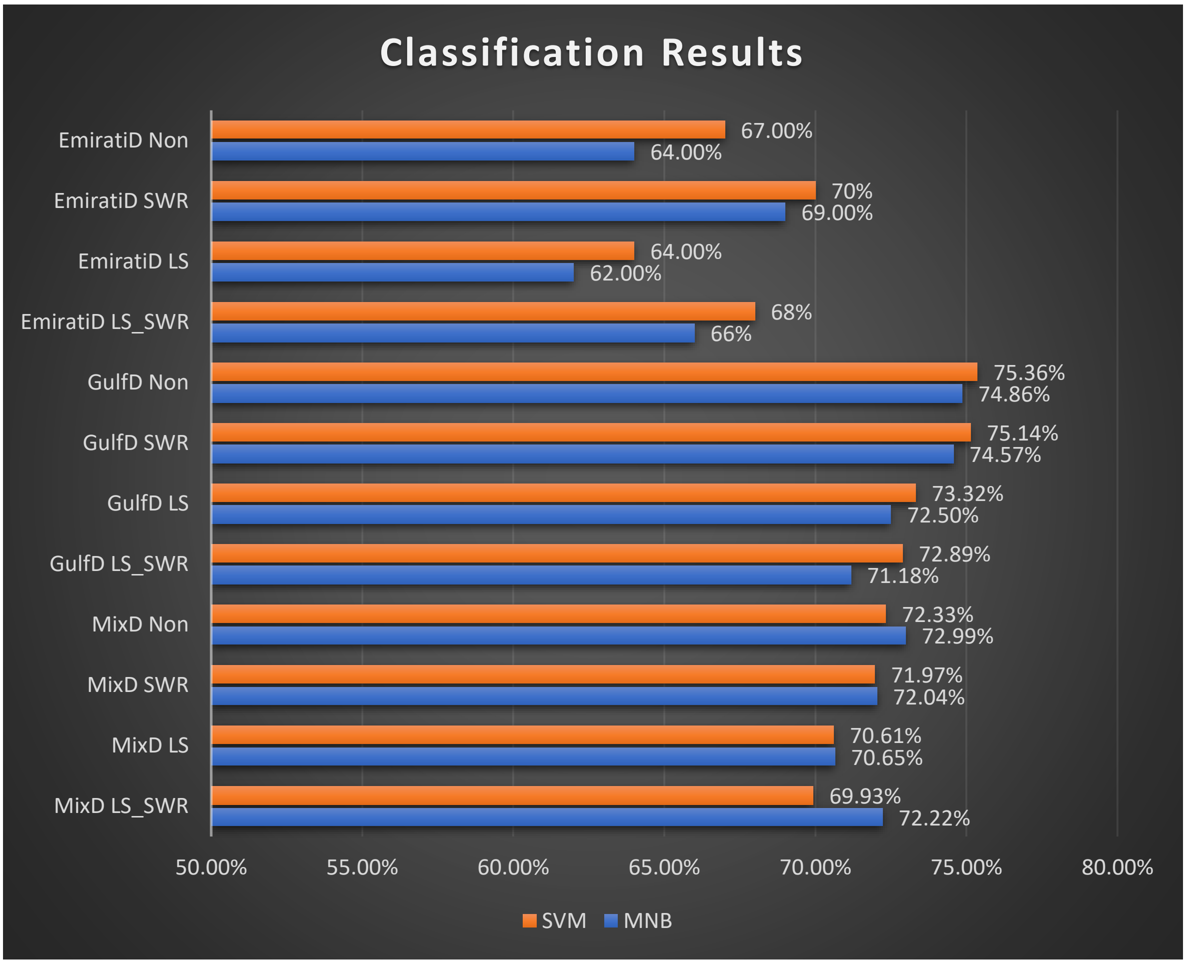


***Figure 4**: visual representation of the classification results.*

Non: no stemming, nor stop words removal used. **SWR**: only stop words removal used
LS: only light stemmer used. **LS_SWR**: both stemming, and stop words removal used.

We can also see the performance of the SVM is better than NB expect for the mix dialect group where the NB showed better results.

4.4.1 Results of Mix Dialect Dataset

Eight experiments are done on mix dialect dataset; 4 by SVM model, and 4 by NB model. The best performance was obtained by NB with accuracy score of 72.99, when no stemming nor stop words removal was applied. We can notice that under these settings, SVM is very close by 0.66%. We also can observe that with mix dialect dataset, NB classifier performed higher regardless of the setting adjustments, than the SVM classifier. The visual representation of all results is shown in figure 5. We can observe that in all different settings with stop words removal, the results are better when no stemmer is used. Also, the NB performed well when both stemmer and Stopwords removal were used. While SVM performed worst when both stemmer and Stopwords removal steps were used. We notice that the performance of both models SVM and NB are close in all scenarios except of the last one (with stemming, and with Stopwords removal). Where NB performed a lot better than SVM. And final observation is regarding the use of stemming; both classifiers didn't perform well when only the stemming is used.

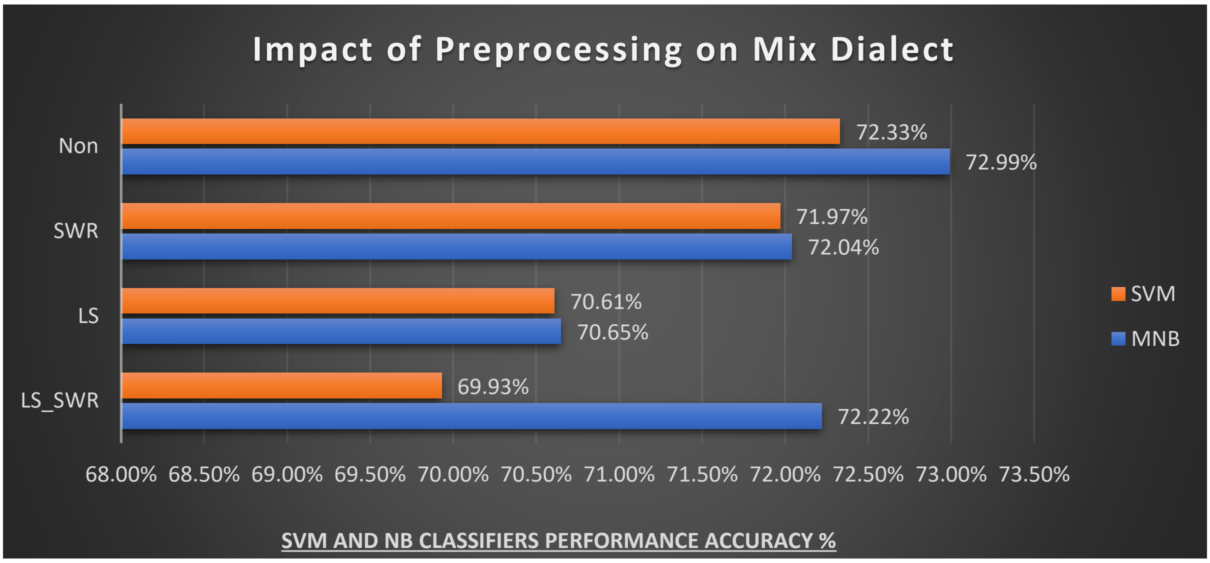


***Figure 5**: results of mix dialect datasets.*

Non: no stemming, nor stop words removal used. **SWR**: only stop words removal used
LS: only light stemmer used. **LS_SWR**: both stemming, and stop words removal used.

4.4.2 Results of Gulf Dialects Dataset

Eight experiments are done on gulf dialect dataset; 4 by SVM model, and 4 by NB model. The best performance was obtained by SVM with accuracy score of 75.36, when no stemming nor stop words removal were applied. We also can observe that with gulf dialect dataset, SVM classifier performed higher regardless of the setting adjustments, than the NB classifier. The visual representation of all results is shown in figure 6. We can clearly see the application of stemming lowering the performance. While, the use of Stopwords removal affected the accuracy of the performance only slightly for both models.

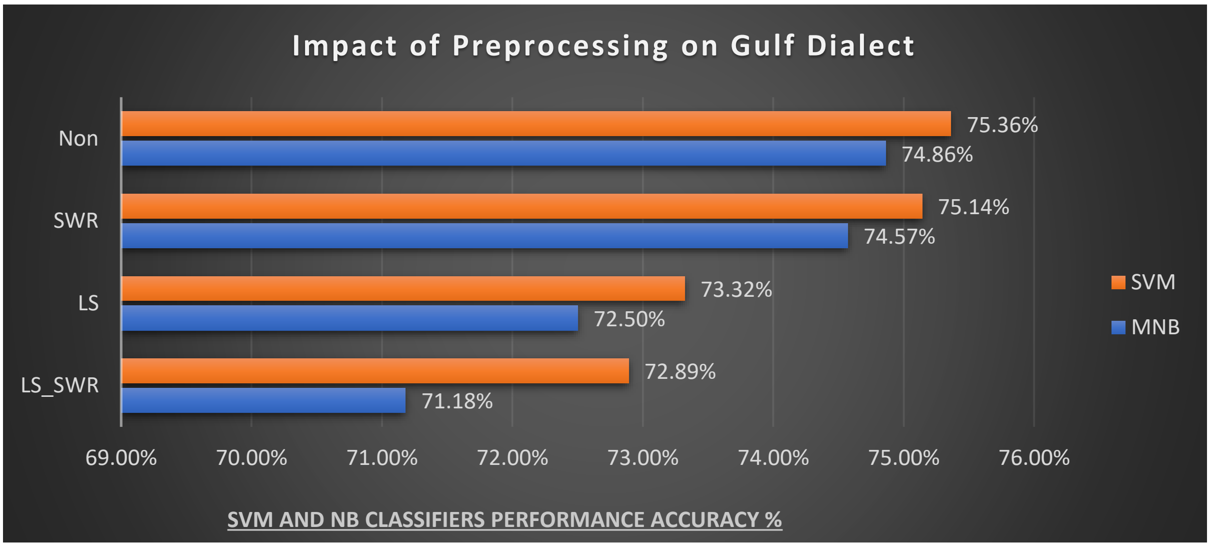


***Figure 6**: Results of gulf dialect datasets*

Non: no stemming, nor stop words removal used. **SWR**: only stop words removal used
LS: only light stemmer used. **LS_SWR**: both stemming, and stop words removal used

4.4.3 Results of Emirati Dialects Dataset

Eight experiments are done on Emirati dialect dataset; 4 by SVM model, and 4 by NB model. The best performance was obtained by SVM with accuracy score of 70%, when no stemming is used, and Stopwords removal is applied. We also can observe that with Emirati dialect dataset, SVM classifier performed higher regardless of the setting adjustments, than the NB classifier. In all scenarios, this difference is significant. The visual representation of all results is shown in figure 7. We can also see the use of stemming alone lowered the performance. While, the use of Stopwords removal enhanced the accuracy of the performance only slightly for both models.

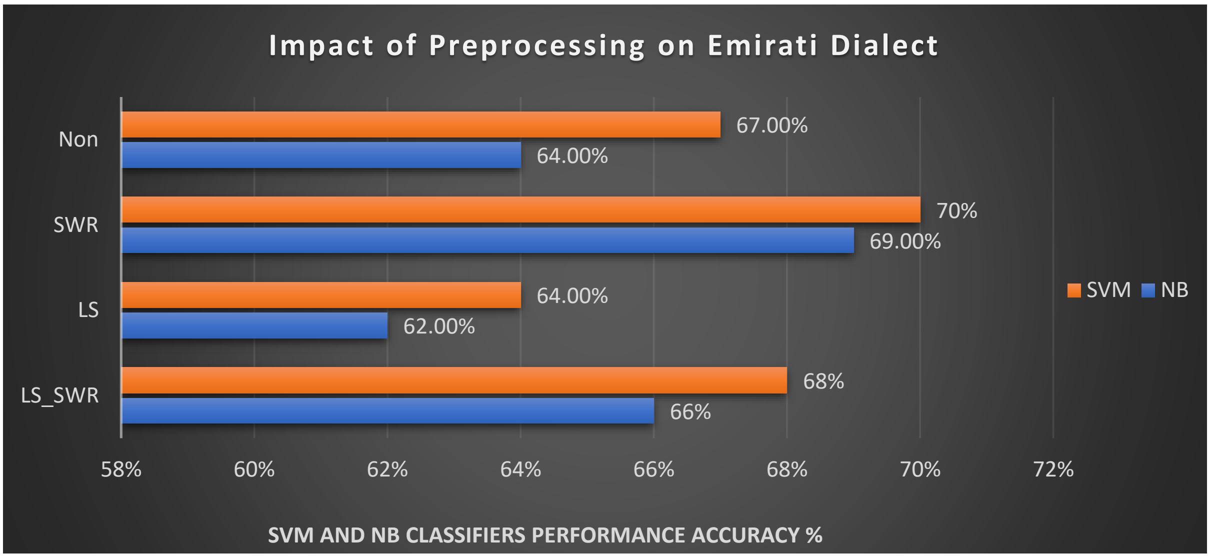


***Figure 7**: results of Emirati dialect sets*

Non: no stemming, nor stop words removal used. **SWR**: only stop words removal used
LS: only light stemmer used. **LS_SWR**: both stemming, and stop words removal used

4.4.4 Results Comparison

When comparing the results as presented in figure 4, the best scenario is performed on gulf dialect dataset with (no stemming, no stop words removal) settings, by SVM classifier. This best scenario scored 75.38% in accuracy measurement. We observe that the top best four scenario as all performed on gulf dialect datasets by both SVM and NB, when no stemming is used. The stemming application in the gulf dialect experiments, did lower the accuracy however not as dramatic as in the other dataset's experiments. However, whether dramatic or not, the stemming did lower the performance in most cases. Another observation is regarding the Stopwords, in mix dialects dataset cases and in gulf dialect dataset the use of Stopwords removal slightly lowered the performance. However, this is the opposite with Emirati dialect, where the best scenario is when Stopwords removal step is added. Overall, both models SVM and NB performed well on gulf dialect dataset, while both models SVM and NB performed the least on the Emirati dialect.

5 Discussion

This experimentational research is conducted to contribute toward Dialectal Arabic sentiment analysis, in answering four research questions stated in section 1.3. We will discuss each question in this section as per the experiments results and our thoughts.

5.1.1 RQ1 which machine learning classifier performs better on dialectal Arabic.

In our experiment we used two classifiers; SVM and NB. Both are known to attain great results in previous and other researches experiments as discussed in section 2. This is also reflected in our study. SVM outperformed NB in most scenarios (eight out of twelve). However, among other four where NB outperformed SVM, the results were two closes in two scenarios. For these reasons, we can confirm that SVM is a better classifier when working with dialectal Arabic dataset, specialy with gulf dialects dataset.

5.1.2 RQ2 what is the effect of different preprocessing and different experiment setting on the classification results.

To answer this question, we studied the impact of two popular preprocessing tasks. The first one is the use of stemming. The second is the task of stop words removal.

First insight is that the performance is getting low with the use of stemming. Stemming is expected to enhance the performance. None of our 24 results reflected any enhancements with stemming. One reason this might be is that our choice of stemmer tools might not be right. We choose Tashaphyne as our light Arabic stemmer tool. Finding the right stemmer is one of the know Arabic sentiment challenges. Our findings are similar to Oussous, Lahcen and Belfkih (2019) in regards to 1) the stemmers don't improve dialectal Arabic, and 2) Tashaphyne doesn't perform well on dialectal Arabic.

Second insights are regarding the use of stop words removal. Stop words removal did enhance the performance in cases of mix dialects and gulf dialects datasets, in contrast to Emirati dialect which dramatically enhanced for both models; SVM and NB. This is because we added some known stopwords from Emirati dialects that are widely used, and because the Emirati dialect dataset is small (500 tweets), in comparison to mix dialect dataset (13982) and gulf dialect dataset (1400), it helps in improving the quality of the dataset. Thus, enhance the models performances.

When both; stemming and stop words removal are used, the performance is enhanced than when only stemmer is used, but lower than when only stopwords removed. This only indicate that the performance of using both stemming and removing stop words, is enhanced because of the stopwords removal not stemmer.

We believe that the quality of stop words could be enhanced, and thus the classification models would be enhanced. Similarly with stemmer. Current stemmers are required to be evaluated further on dialectal Arabic and specially gulf dialects, to find the best stemmer to be used in gulf dialect sentiment analysis projects.

5.1.3 RQ3 does the same ML model performs better on one Arabic dialects group, in comparison to mix Arabic dialects

This is an important and our core question. From our experiment, we can clearly state that the Arabic dialects group should be considered when performing dialectal sentiment analysis. Both machine learning classifiers (SVM and NB), performed better with gulf dialect group dataset that its performance on mix dialect groups under same settings, in all eight scenarios. From this insight, we believe more efforts should be put into dialectal Arabic, one group dialects such as; Egyptian dialect, gulf dialect, levant dialect, Iraqi dialect, and north Africa dialects.

5.1.4 RQ4 does the same ML model performs better on one dialect, in comparison to mix dialects

From this experiment, as we the results of both models are noticeably low, the concentration should be made on one dialect groups. However, to be able to clearly answer this question, more investigation required on the case of low results with Emirati dialects. There are some limitations with our experiments when the models applied to the Emirati dialects dataset. The main one is the size of the dataset. The Emirati dialects datasets we used has only 500 records, while the other datasets have around 14000 each.

We believe performing sentiment analysis on one dialect group is enough while considering the differences in the dialects of one group. With the globalization and mix cultures, many people speak with mix dialects, just as how they speak with mix languages. Many Emirati citizen are originally from other gulf countries such as Saudi, or Yamen. So most of those people would use mix dialects when they communicate. In fact, the words we selected are widely use in UAE by its citizen, however if we need to expand the dataset, we will end up using same words and phrases that are used by other gulf countries, because there are more similarities than differences in the different gulf dialects.

6 Future work

Sentiment analysis is a very effective tool, yet it is very challenging when dealing with dialectal Arabic text. This research paper is on an early stage of gulf dialects and Emirati dialect sentiment analysis. More investigations needs to be conducted in this area. Currently, to our knowledge, not enough resources on the targeted dialects are available.

We consider this work as the base for many enhancements and experiments in the future. There are many ways we would experiments this study in the future. We would like to attempt two approaches;

1. Hybrid approach:
   By developing an Emirati – or gulf - lexicons to feed the learning process.
2. Deep learning approach:
   Long Short-Term Memory (LSTM) and GRU (gated recurrent neural networks unit). As seen in the related work, good results attained by LSTM and GRU deep learning approach.

Further more the multiple stemmers need to be evaluated against gulf dialects for find the best stemmers of gulf dialects. Additionally, stopwords list should be evaluated and enhanced to fit the gulf dialects. And finally, we will use a larger dataset for Emirati dialects, to evaluate the classification performance and Emirati dialect sentiment analysis.

7 Conclusion

Sentiment analysis is popular Natural Language Processing field, today. The main task of sentiment analysis is to extract useful insights (sentiments) from a given text data, through sentiment polarity classification. Sentiment analysis is a promising tool that can utilized in many fields.

This research paper presented an overview understanding of the Arabic sentiment analysis. We shed lights on; the importance of the topic and challenges of Arabic language sentiment analysis. The paper also discussed some of the recent important experimentational researches on the field. The next part of the research attempted to answer our research questions through a practical experiment.

This paper contributed toward the Arabic dialects' sentiment analysis. It examined the effect of different Arabic dialect levels on the performance of the classification. To our best knowledge, no work has been done on sentiment analysis that compare the performance of classification based on three Arabic levels, starting from general mix Arabic dialects to one specific Arabic dialect. This paper also contributing to the sentiment analysis on gulf dialects.

Furthermore, we evaluated two machine learning classifiers performances on sentiment analysis. The fist one is the popular Support Vector Machine (SVM). The second classifier is Naïve Bayes (NB). Out of twelve experiments implemented, SVM attained higher accuracy in eight experiments, while NB achieved better accuracy in four experiments.

We conducted the experiment using machine learning approach. Three datasets were prepared. The first one the is dataset with mix Arabic dialects. The second dataset is dataset with only gulf dialects, which include mix dialects of gulf dialect group. The third dataset is the Emirati dialect dataset. Next, we performed preprocessing when included; normalization, data cleansing, tokenization, stemming, and stop words removal. The feature extraction method that is used is TF-IDF.

We conducted total of 12 experiments for each classifier; SVM and NB, on three relevant datasets, with different settings. We also evaluated the performance of two machine learning classifier on sentiment analysis for Arabic dialects. The best accuracy results was 75.38% by SVM classifier run on gulf dialect dataset with the following setting; no stemmer and no Stop words removal.

The results show sentiment analysis on one dialects group attain more accurate results, in comparison to the mix dialect dataset. While the question related to one dialect is difficult to answer due to the data size in comparison to the other two datasets, we recommended that in sentiment analysis gulf dialect group should be considered rather than one specific dialect.

On the technicality of the sentiment analysis, the research also studies the impact of the preprocessing on Arabic sentiment analysis, on term of stemming and stop words removal. More studies is recommended to be conducted on gulf dialects sentiment analysis.

References

1) Abdullah, M., AlMasawa, M., Makki, I., Alsolmi, M., & Mahrous, S. (2018). Emotions extraction from Arabic tweets. *International Journal of Computers and Applications*, *42*(7), 661-675.

2) Abo, M. E. M., Raj, R. G., & Qazi, A. (2019). A Review on Arabic Sentiment Analysis: State-of-the-Art, Taxonomy and Open Research Challenges. *IEEE Access*, *7*, 162008-162024.

3) Alayba, A. M., Palade, V., England, M., & Iqbal, R. (2017, April). Arabic language sentiment analysis on health services. In *2017 1st international workshop on arabic script analysis and recognition (asar)* (pp. 114-118). IEEE.

4) Al-Ayyoub, M., Khamaiseh, A. A., Jararweh, Y., & Al-Kabi, M. N. (2019). A comprehensive survey of arabic sentiment analysis. *Information processing & management*, 56(2), 320-342.

5) Al-Harbi, O. (2019). Classifying sentiment of dialectal arabic reviews: a semi-supervised approach. Int. Arab J. Inf. Technol., 16(6), 995-1002.

6) Almuqren, L. A., Qasem, M. M., & Cristea, A. I. (2019). Using deep learning networks to predict telecom company customer satisfaction based on Arabic tweets.

7) Alomari, K. M., ElSherif, H. M., & Shaalan, K. (2017, June). Arabic tweets sentimental analysis using machine learning. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 602-610). Springer, Cham.

8) Alruily, M. (2020). Issues of dialectal saudi twitter corpus. Int. Arab J. Inf. Technol., 17(3), 367-374.

9) Areed, S., Alqaryouti, O., Siyam, B., & Shaalan, K. (2020). Aspect-based sentiment analysis for Arabic government reviews. In *Recent Advances in NLP: The Case of Arabic Language* (pp. 143-162). Springer, Cham.

10) Al-Smadi, M., Talafha, B., Al-Ayyoub, M., & Jararweh, Y. (2019). Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. *International Journal of Machine Learning and Cybernetics*, 10(8), 2163-2175.

11) Al-Saqqa, S., Obeid, N., & Awajan, A. (2018, October). Sentiment analysis for Arabic text using ensemble learning. In *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)* (pp. 1-7). IEEE.

12) Al Suwaidi, H., Soomro, T. R., & Shaalan, K. (2016). Sentiment analysis for emiriti dialects in twitter. *Sindh University Research Journal-SURJ* (Science Series), 48(4).

13) Al-Twairesh, N., Al-Khalifa, H., Alsalman, A., & Al-Ohali, Y. (2018). Sentiment analysis of arabic tweets: Feature engineering and a hybrid approach. *arXiv preprint arXiv:1805.08533*.

14) Al-Thubaity, A., Alqahtani, Q., & Aljandal, A. (2018). Sentiment lexicon for sentiment analysis of Saudi dialect tweets. *Procedia computer science*, 142, 301-307.

15) Atoum, J. O., & Nouman, M. (2019). Sentiment analysis of Arabic jordanian dialect tweets. *Int. J. Adv. Comput. Sci. Appl*, *10*(2), 256-262.

16) Baali, M., & Ghneim, N. (2019). Emotion analysis of Arabic tweets using deep learning approach. *Journal of Big Data*, *6*(1), 1-12.

17) Baly, R., Khaddaj, A., Hajj, H., El-Hajj, W., & Shaban, K. B. (2019). Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in arabic levantine tweets. arXiv preprint arXiv:1906.01830.

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

19) DoniaGamal, M. A., El-Horbaty, E. S. M., & Salem, A. B. (2018). Opinion Mining for Arabic Dialects on Twitter. Egyptian Computer Science Journal, 42(4).

20) El-Alfy, E. S. M., & Al-Azani, S. (2020). Empirical study on imbalanced learning of Arabic sentiment polarity with neural word embedding. *Journal of Intelligent & Fuzzy Systems* (1064- 1246), 1-12.

21) Farha, I. A., & Magdy, W. (2019). Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop* (pp. 192-198).

22) Ghallab, A., Mohsen, A., & Ali, Y. (2020). Arabic sentiment analysis: A systematic literature review. Applied Computational Intelligence and Soft Computing, 2020.

23) Gwad, W. H. G., Ismael, I. M. I., & Gültepe, Y. (2020). Twitter Sentiment Analysis Classification in the Arabic Language using Long Short-Term Memory Neural Networks. *International Journal of Engineering and Advanced Technology (IJEAT), 9 (3), (*2249 – 8958).

24) Hammad, M., & Al-awadi, M. (2016). Sentiment analysis for arabic reviews in social networks using machine learning. In *Information technology: new generations* (pp. 131-139). Springer, Cham.

25) Hegazi, M. O., Al-Dossari, Y., Al-Yahy, A., Al-Sumari, A., & Hilal, A. (2021). Preprocessing Arabic text on social media. *Heliyon*, 7(2), e06191.

26) Jerbi, M. A., Achour, H., & Souissi, E. (2019). Sentiment analysis of code-switched tunisian dialect: Exploring RNN-based techniques. In *International Conference on Arabic Language Processing* (pp. 122-131). Springer, Cham.

27) Johnson, J. (2021). Most common languages used on the internet. [online]. [Accessed 11 April 2021] Available at: https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/

28) Kumar, S., & Zymbler, M. (2019). A machine learning approach to analyze customer satisfaction from airline tweets. *Journal of Big Data*, *6*(1), 1-16.

29) Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies,* 5(1), 1-167.

30) Manguri, K. H., Ramadhan, R. N., & Amin, P. R. M. (2020). Twitter sentiment analysis on worldwide COVID-19 outbreaks. *Kurdistan Journal of Applied Research*, 54-65.

31) Nejjari, M., & Meziane, A. (2019). Overview of Opinion Detection Approaches in Arabic. In *Proceedings of the 2nd International Conference on Networking, Information Systems & Security* (pp. 1-5).

32) Oussous, A., Lahcen, A. A., & Belfkih, S. (2019, March). Impact of text pre-processing and ensemble learning on Arabic sentiment analysis. In Proceedings of the 2nd International Conference on Networking, Information Systems & Security (pp. 1-9).

33) Oueslati, O., Cambria, E., HajHmida, M. B., & Ounelli, H. (2020). A review of sentiment analysis research in Arabic language. *Future Generation Computer Systems*, *112*, 408-430.

34) Salloum, S. A., AlHamad, A. Q., Al-Emran, M., & Shaalan, K. (2018). A survey of Arabic text mining. In *Intelligent Natural Language Processing: Trends and Applications* (pp. 417-431). Springer, Cham.

35) Shaalan, K., Siddiqui, S., Alkhatib, M., & Monem, A. A. (2019). Challenges in Arabic natural language processing. *Computational Linguistics*.

36) Soufan, A. (2019). Deep learning for sentiment analysis of arabic text. In *Proceedings of the ArabWIC 6th Annual International Conference Research Track* (pp. 1-8).

37) Szmigiera, M. (2021). The most spoken languages worldwide in 2021. [online]. [Accessed 11 April 2021] Available at: https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/

38) Tankovska, H. (2021). Global social networks ranked by number of users. [online]. [Accessed 5 March 2021] Available at: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

39) United Nations. Official Languages. [online]. [Accessed 11 April 2021] Available at: https://www.un.org/en/our-work/official-languages

40) Wikipedia. (2021). *Arabic*. [online]. [Accessed 13 April 2021] Available at: https://en.wikipedia.org/wiki/Arabic

41) Zahidi, Y., El Younoussi, Y., & Al-Amrani, Y. (2021). Different valuable tools for Arabic sentiment analysis: a comparative evaluation. *International Journal of Electrical & Computer Engineering, 11*(1), 2088-8708.

42) Zerrouki, T. (2019). Tashaphyne Arabic Light Stemmer. [online]. [Accessed 20 January 2021] Available at: https://pypi.org/project/Tashaphyne/

Appendixes

Appendix A: Summary of Related Words

Authors	Year	Task	Datasets	Preprocessing	Features	Approach	Classifiers	Results
Oueslati et al.	2020	sentiment analysis on Arabic reviews	Hotel Arabic reviews dataset (HARD)	-	-	ML	RF, NB, SVM	Definition of the key characteristics of the Arabic sentiment analysis process
El-Alfy and Al-Azani	2020	9 ML algorithms comparison with use of word embedding 6 oversampling techniques (Syria dialect)	1798 tweets	Stopwords removal Normalization Stemming	word2vec	ML	SVM GNB SGD NN DT RF GB VE SE	SGD (with oversampling technique) showed best results in GM (Geometric Mean) measurement.
Areed et al. (2020)	2020	Aspect based sentiment analysis on clients feedback on UAE government mobile applications	2000	Normalization Noise Cleaning	-	Lexicon-based and rules-based	-	this approach passed the baseline results by 6% in accuracy and 17% in F-measure
Gwad, Ismael and Gültepe	2020	Comparing machine learning and deep learning models on Arabic Twitter sentiment analysis	2000 tweets	-	Word2Vec	ML DL	SVM NB K-NN D-Tree LSTM	LSTM showed the highest results (89.8%) followed by SVM (84.7%)
Manguri, Ramadhan and Amin (2020)	2020	Arabic sentiment analysis application on Covid-19	530232 tweets	Normalization	-	ML	NB	Able to classify successfully identify the emotion of people toward coronvirus

Gamal et al.	2019	Methodology to construct Arabic twitter dataset (MSA & Egyptian dialect)	151,000 tweets	Remove Non-Arabic letters Stopwords removal Elongation removal Letter normalization	TF-IDF	ML	SVM NB RP ME AdaBoost	RP yielded the highest accuracy of 99.9%
Baly et al.	2019	Present Levantine dialect twitter dataset (multi-topic and target-based sentiment analysis)	4000 tweets	Remove Non-Arabic letters Cleaning Emoticon removal	Unigram and Bigram TF-IDF	ML	SVM LR RF Trees Ridge	LR showed the best results
Al-Samadi et al.	2019	Aspect based sentiment analysis using LSTM	24,028 hotels reviews		OTEs	DL	LSTM	The results outperformed the baseline research
Farha and Magdy (2019)	2019	Presenting online tool for Arabic SA	9655 tweets	Letter normalization Elongation removal Cleaning	Word embeddings	DL	CNN LSTM	State-of-the art results
Oussous, Lahcen and Belfkih (2019)	2019	Exploring the impact of pre-processing stage	40,000 Tweets and reviews	Normalization Stopwords removal Remove Non-Arabic letters Remove duplicate tweets Removal elongation Noise cleansing	Stem N-gram Stopwords	ML	SVM NB ME	SVM outperformed other classifiers in all cases. Better results with light stemming, unigram, and with no filtration of Stopwords.
Al-Harbi (2019)	2019	Sentiment analysis using semi-Supervised approach on Jordanian dialect	2500 reviews	Correcting misspelling Removing Elongation removal Letter normalization Emoticons removed	PWN, NWN, NgWN, CWA, PCP, NCP, PWP, NWP, RL	Hybrid approach (semi-supervised)	SVM NB RF K-NN	SVM showed best performance with accuracy = 92.3%
Almuqren, Qasem and Cristea (2019)	2019	Predict customer satisfaction of Saudi telecommunication companies	20,000 tweets	Remove Non-Arabic letters Cleaning Normalization	n-gram TF-IDF Is-Sarcastic' feature Affective-cue Tweet topic	ML DL	SVM LSTM GRU	The best classifier is bidirectional-GRU with attention mechanism

Baali and Ghneim (2019)	2019	Emotion analysis using/comparing ML & DL approaches	5600 tweets	Letter Normalization Stopwords removal Cleaning Stemming	Stem TF-IDF Count Vectorizer	DL ML	CNN SVM NB MLP	CNN outperformed ML classifiers (99.9% accuracy on training and 99.8 on validation)
Kaity and Balakrishnan (2019)	2019	Built framework to automatically generate Arabic sentiment lexicon.	10,219 documents (Facebook data)	Stopwords removal Remove non-Arabic letters Cleaning Lemmatizing		Lexicon-based approach	-	0.74 F measure
DoniaGamal et al.	2018	Dialectal Arabic sentiment analysis	151500 tweets	Letters Normalization Cleaning	TF	ML	SVM NV MNB BNB LR SGD	SVM showed best accuracy score of 93.56% Followed by LR 93.52%
Goel and Thareja	2018	Emotion analysis using twitter hashtags (emotion keywords)	4000 tweets			Lexicon-based approach (emotion lexicon)	-	Hashtags can be used to enhance twitter emotion analysis
Hammad and Al-awadi	2018	Finding the best “lightweight” approach to sentiment analysis	2000 reviews (Facebook, YouTube, Twitter)	Light Stemming Root Stemming Stopwords removal Cleaning		ML DL	SVM NB DT BPNN	SVM showed best score in F measurement SVM also was best in learning with training data size increase SVM took shortest to train data
Siddiqui, Monem and Shaalan	2018	Enhancing Arabic lexicon sentiment analysis approach	2000 tweets 500 film reviews	-	-	Rules-based approach (heuristics rules)	-	93.9 accuracy on twitter data, and 85.6% accuracy on the OCA data. There is increase by 23.85% in accuracy in comparison with the baseline.

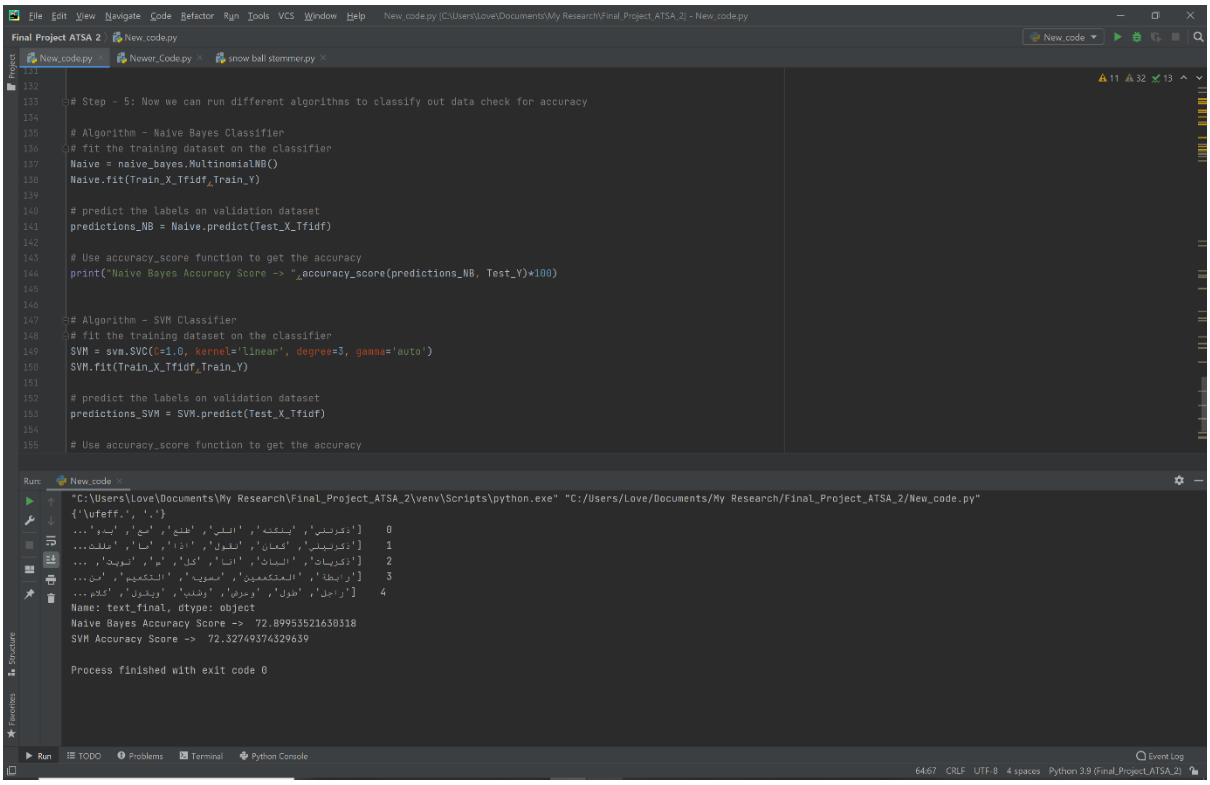
Abdulla et al.	2018	Proposed new emotion classification model AR_EMC	2025 tweets	Stemming Stopwords removal Cleaning	Unigram, and Bigram TF-IDF	ML	SVM, NB, J48, SMO	SVM (best accuracy = 80.6%) NB (best ROC =0.95%)
Alomari, ElSherif and Shaalan	2017	Introducing Jordanian Arabic corpus for sentiment analysis	1800 tweets	Noise cleansing Remove Non-Arabic letters Tokenization Stemming Word’s filtration	TF-IDF Stemming N-garam	ML	SVM NB	SVM with stemming, using bigrams, and TF-IDF features, showed best results (88.72%)
Alayba et al.	2017	Arabic sentiment analysis application on health services tweets	2026 tweets	Normalization Noise cleansing Removed neutral tweets Removed spam, duplicate, and retweets, Removed unrelated to health tweets	TF-IDF Unigram and Bigram	ML DL	SVM MNB LR CNN	SVM classifiers showed best results using LSV and SGD.
Mdhaffar et al.	2017	Sentiment Analysis of Tunisian Dialect	17,000 Facebook comments	Noise cleansing Remove non-Arabic comments	-	ML	SVM NB MLP	MLP classifiers showed best results. Tunisian datasets showed better result in comparison to other datasets.
Al Suwaidi et al.	2016	Sentiment Analysis for Emirati Dialects in Twitter	1000 tweets	-	-	-	-	Emirati words were successfully labelled.

Appendix B: Stopwords

```
عجبا', 'فيها', 'فكان', 'بن', 'أيلول', 'ذِه', 'بعض', 'أنّى', 'علم', 'أ', 'فإن', 'انها', 'لوكالة', 'تعلّم', 'الثانية', 'كأن', 'هؤلاء', 'ئ', 'ثلاثين', '{
'بسّ', 'ءَ', 'يونيو', 'يمين', 'لام', 'مالنفك', 'وهي', 'ؤ', 'منذ', 'ألفى', 'س', 'ذيت', 'ان', 'لين', 'تاء', 'آهاً', 'سرعان', 'الاولى', 'اللواتي',
'لذلك', 'تحت', 'تِه', 'الان', 'ثالث', 'عام', 'في', 'الا', 'بهذا', 'خاء', 'هَجْ', 'مئة', 'ما', 'أيّان', 'إلّا', 'بضع', 'دونك', 'ايار', 'حين', 'هيّا',
'حول', 'صار', 'ه', 'دون', 'خمسة', 'التى', 'عين', 'آ', 'مايزال', 'مادام', 'ذَيْنِ', 'عنها', 'أمد', 'ذات', 'واهاً', 'لوما', 'لها', 'سحقا',
'أربعمئة', 'اللذان', 'امس', 'كانون', 'وقد', 'أكتوبر', 'الألى', 'انقلب', 'ثمَّ', 'ضد', 'إليكنّ', 'إنك', 'إياك', 'سيما', 'ستكون', 'ارتدّ', 'ثمانية',
'إزاء', 'اللتين', 'طاق', 'أخذ', 'فى', 'ليت', 'ترك', 'قوة', 'بد', 'عندما', 'إياكما', 'أصلا', 'الف', 'عشرة', 'ليس', 'أوّهْ', 'ماذا', 'كى', 'إذما',
'أنتم', 'وراءَك', 'تانِك', 'تبدّل', 'ثلاثة', 'صبرا', 'عاشر', 'ولايزال', 'سوى', 'ق', 'مقابل', 'هلم', 'كأيّ', 'أولئك', 'ثلاث', 'رابع', 'طاء',
'اصبح', 'برح', 'كما', 'سبعة', 'نون', 'إياكم', 'اربعة', 'بئس', 'أنتِ', 'مما', 'إن', 'ذانِ', 'حار', 'له', 'أمس', 'ا?ى', 'أبٌ', 'اعادة', 'رزق',
'عاما', 'ترا', 'ذهب', 'إلي', 'مو', 'تموز', 'ومع', 'الاول', 'صاد', 'هَاتِه', 'تكون', 'أين', 'مليون', 'هناك', 'بَلْهَ', 'او', 'عل', 'نا', 'فاء',
'أنشأ', 'وقال', 'حمٌ', 'ب', 'وُشْكَانَ', 'سنوات', 'تسعة', 'أمام', 'وله', 'معاذ', 'إذاً', 'ابي', 'شبه', 'إذا', 'خمسمئة', 'ومن', 'تلقاء', 'الآن',
'ذانك', 'و', 'يكون', 'تجاه', 'والتي', 'يوم', 'بس', 'كاد', 'بإن', 'وثي', 'ورد', 'أنتن', 'إياهن', 'لدي', 'أرى', 'مذ', 'اني', 'لهم', 'إليكَ',
'حجا', 'آنفا', 'مكانَك', 'كانت', 'قال', 'شين', 'آهِ', 'التي', 'كسا', 'ساء', 'ظاء', 'عنه', 'اتخذ', 'بغتة', 'ذا', 'اليوم', 'لكنه', 'حيثما', 'كأيّن',
'حمدا', 'سنة', 'أولالك', 'ظنَّ', 'عَدَسْ', 'صراحة', 'مازال', 'أربعة', 'هَذِي', 'أيضا', 'ـ', 'ذِي', 'اذا', 'أي', 'جي', 'إنَّ', 'فوق', 'مليم', 'ل',
'ضحوة', 'علق', 'حبيب', 'انا', 'ستون', 'قبل', 'قاف', 'لقاء', 'ايضا', 'عاد', 'بخٍ', 'عيانا', 'هيا', 'لو', 'نَخْ', 'إى', 'ظل', 'أحد', 'أربعمائة',
'لس', 'االتى', 'امسى', 'سبتمبر', 'ريث', 'صبر', 'أم', 'أجل', 'اكد', 'إحدى', 'الي', 'يا', 'أنتما', 'أيار', 'بدلا', 'ولا', 'أخٌ', 'به', 'هاكَ',
'عشر', 'انه', 'أربعاء', 'تسعمائة', 'اثنين', 'تِي', 'المقبل', 'يناير', 'هاي', 'بات', 'أخبر', 'حتى', 'ستمائة', 'إيهٍ', 'بلى', 'تارة', 'ء', 'إياي',
'بطآن', 'حادي', 'فيج', 'وبين', 'خ', 'ثاني', 'لأن', 'أفعل', 'إمّا', 'اخلولق', 'دينار', 'شَتَّانَ', 'لي', 'أطعم', 'كلتا', 'سوف', 'كن', 'اثنا',
'وفي', 'إياها', 'خامس', 'جير', 'آض', 'عشرون', 'اللذين', 'قطّ', 'مايو', 'تسعمئة', 'ألا', 'حاء', 'فيه', 'ثم', 'ح', 'كلّما', 'ثمان', 'فيفري',
'هَاتَيْنِ', 'آه', 'جعل', 'لنا', 'بسبب', 'خمس', 'هم', 'صباح', 'نحن', 'لعل', 'تم', 'أفٍّ', 'هللة', 'غادر', 'إياهما', 'ا', 'هيهات', 'جمعة',
'مكانكم', 'قام', 'واو', 'تَيْنِ', 'چذي', 'أعلم', 'أنت', 'رويدك', 'ليسب', 'كِخ', 'كلَّا', 'بعدا', 'تاسع', 'ذو', 'طرا', 'الاخيرة', 'االا', 'حاليا',
'ابتدأ', 'مع', 'ويش', 'اليه', 'عسى', 'أقبل', 'ا?', 'مارس', 'إذ', 'خمسمائة', 'تسع', 'فو', 'هَذِه', 'آب', 'بها', 'لما', 'نوفمبر', 'وكان', 'سمعا',
'اخرى', 'إذن', 'عنس', 'ى', 'لكن', 'يمكن', 'خمسين', 'لا', 'وان', 'زي', 'أمامك', 'ثلاثون', 'ها', 'حرى', 'ج', 'ظ', 'يوليو', 'عند', 'مثل',
'مائة', 'ولم', 'هنالك', 'أمامكَ', 'حيَّ', 'جميع', 'يفعلان', 'نَّ', 'ترى', 'واضافت', 'ثماني', 'متى', 'وكانت', 'آذار', 'أن', 'عشرين', 'اثني',
'الحالي', 'ع', 'ثمانمئة', 'تلك', 'يورو', 'ذولي', 'كأنّ', 'حزيران', 'عامة', 'ستة', 'مكانكنّ', 'ثلاثمئة', 'واضاف', 'فرادى', 'لعمر',
'أغسطس', 'ن', 'واكد', 'عدد', 'كيت', 'واحد', 'خبَّر', 'وا', 'عندي', 'أولاء', 'جنيه', 'نحو', 'دولار', 'وهو', 'وليس', 'كان', 'خمسون',
'الألاء', 'تفعلون', 'كلكم', 'سادس', 'ظلّ', 'ذ', 'نفسه', 'حوالى', 'ز', 'سنتيم', 'راح', 'أصبح', 'تينك', 'هذا', 'طفق', 'اثنان', 'لات', 'لمّا',
'إليكما', 'انك', 'غير', 'هل', 'هَذانِ', 'ياء', 'اطار', 'ثمانون', 'تسعون', 'عليك', 'الوقت', 'فهو', 'مه', 'أخو', 'قرش', 'كثيرا', 'ولكن',
'وَيْ', 'لاسيما', 'بين', 'زعم', 'خاصة', 'درى', 'تعسا', 'أنبأ', 'حبذا', 'تانِ', 'جوان', 'لم', 'ث', 'اللاتي', 'هبّ', 'أيّان', 'ألف', 'لعلَّ',
'هَاتانِ', 'منها', 'عليها', 'معه', 'وعلى', 'منه', 'هَاتِي', 'ين', 'إيانا', 'أى', 'هنا', 'أفٍّ', 'بشكل', 'ك', 'الذي', 'إلى', 'حاشا', 'أهلا', 'صهٍ',
'السابق', 'إياهم', 'ثمانين', 'حسب', 'غداة', 'سبعين', 'اجل', 'دواليك', 'ش', 'مساء', 'راء', 'انبرى', 'خلال', 'خلف', 'منهو', 'كيفما',
'انت', 'جانفي', 'مهما', 'مرّة', 'اربعين', 'أبريل', 'أمّا', 'الماضي', 'ضمن', 'هما', 'فضلا', 'وهذا', 'أجمع', 'ي', 'أنك', 'تحوّل', 'فقط',
'لن', 'ة', 'ثامن', 'ط', 'غدا', 'هَيْهات', 'لايزال', 'لولا', 'إياكن', 'زيارة', 'جيم', 'آل', 'هَذَيْنِ', 'شخصا', 'كلم', 'أما', 'برس', 'عوض',
'وأن', 'لدى', 'كم', 'اضحى', 'جدا', 'ديسمبر', 'وجد', 'سبحان', 'أول', 'اليها', 'بعد', 'علي', 'لازال', 'ثلاثمائة', 'تفعلان', 'قاطبة', 'حدَث',
'فقد', 'سبت', 'لهذا', 'كذلك', 'مال', 'أبدا', 'شباط', 'لدن', 'تسعين', 'إياه', 'قلما', 'ثمّة', 'كذا', 'أسكن', 'هي', 'و6', 'وقف', 'مئتان', 'كله',
بؤسا', 'آمينَ', 'نهاية', 'هذي', 'لكنَّ', ' ,'"،"\ufeff' ,'بيد', 'غ', 'نيسان', 'ثاء', 'ليرة', 'هَؤلاء', 'اللتان', 'ت', 'إِلَيْكَ', 'د', 'ذينك', 'اي
'فلس', 'مابرح', 'شرع', 'بان', 'جويلية', 'عدة', 'أفعله', 'من', 'اربعون', 'دال', 'هذه', 'ذاك', 'نبَّا', 'سابع', 'أو', 'يلي', 'ست', 'كل', 'كاف',
'كيف', 'ض', 'للامم', 'نفس', 'منو', 'سبع', 'وهب', 'ثمّ', 'ضاد', 'إليكم', 'حمو', 'هو', 'اكثر', 'ميم', 'فبراير', 'هن', 'شيكل', 'ابين',
'زاي', 'صهْ', 'م', 'فانه', 'همزة', 'أنا', 'رأى', 'أربع', 'عليه', 'ستين', 'الذين', 'تفعلين', 'ايام', 'زود', 'ثلاثاء', 'خميس', 'ر', 'حاي',
'مافتئ', 'جلل', 'تخذ', 'عدا', 'والذي', 'ف', 'عدَّ', 'طَق', 'أضحى', 'على', 'وقالت', 'ماهو', 'غالبا', 'ذلك', 'لبيك', 'نيف', 'الثاني', 'عدم',
'عن', 'خال', 'وأبو', 'سين', 'سبعمئة', 'حيث', 'ثان', 'بل', 'يفعلون', 'هَذا', 'خلا', 'اف', 'أوشك', 'بَسْ', 'اثر', 'غين', 'آناء', 'قد', 'فان',
'شمال', 'ذال', 'اللتيا', 'مليار', 'يوان', 'سبعون', 'باء', 'ريال', 'ص', 'حقين', 'باسم', 'أوت', 'هلّا', 'اول', 'امام', 'شتانَ', 'أعطى', 'صدقا',
'فلان', 'انفك', 'أيّ', 'هاء', 'واوضح', 'أل', 'اعلنت', 'الذاتي', 'صفر', 'أفريل', 'اللي', 'تشرين', 'أيا', 'أمسى', 'سبعمائة', 'سقى', 'طالما',
'مكانكما', 'رُبَّ', 'حَذارِ', 'آهٍ', 'اما', 'خلافا', 'استحال', 'نعم', 'ماي', 'أبو', 'حقا', 'الى', 'رجع', 'الذى', 'احد', 'درهم', 'كرب', 'أنَّ',
}''سرا', 'حق', 'أنه', 'ستمئة', 'علَّ', 'ثمنمئة
```

Appendix C: Results from python programming

I. Results of mix dialects dataset (no stemming, no Stopwords removal)

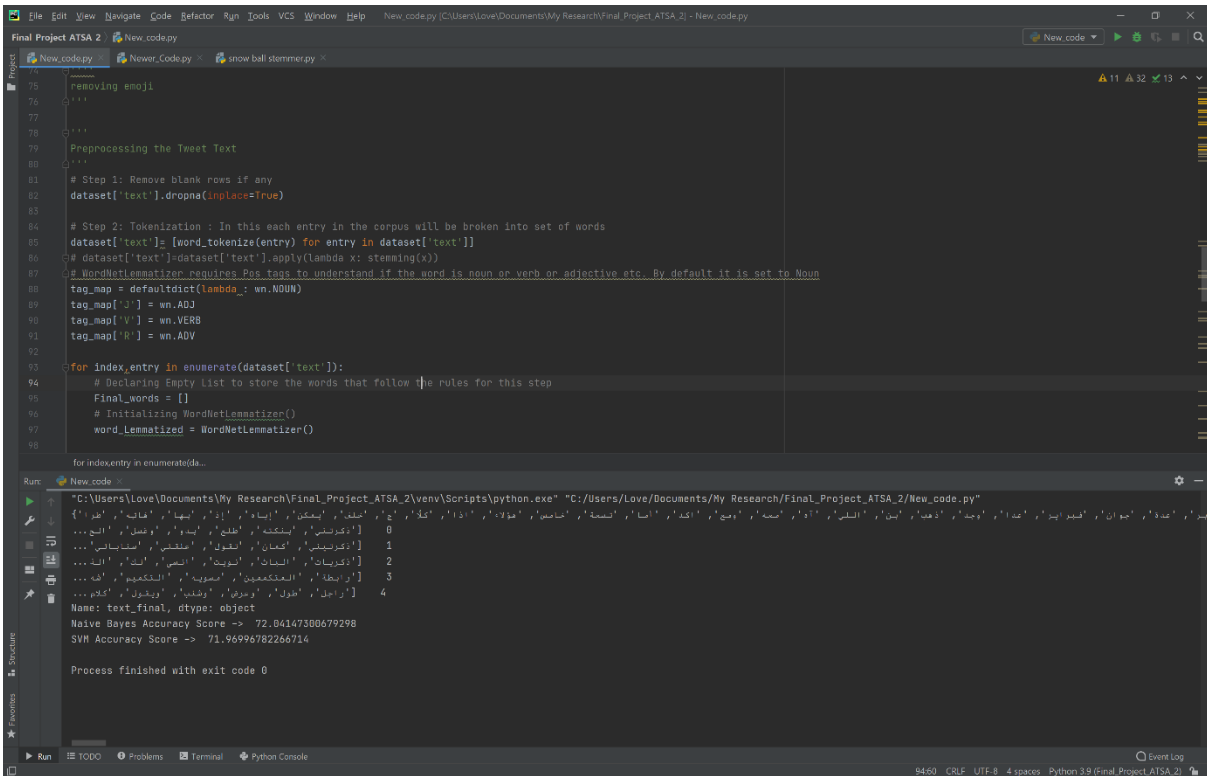


File Edit View Navigate Code Refactor Run Tools VCS Window Help
New_code.py [C:\Users\Love\Documents\My Research\Final_Project_ATSA_2] - New_code.py
Final Project ATSA 2 New_code.py
New_code.py Newer_Code.py snow ball stemmer.py
Run: New_code
Run TODO Problems Terminal Python Console Event Log
64:67 CRLF UTF-8 4 spaces Python 3.9 (Final_Project_ATSA_2)

```
# Step - 5: Now we can run different algorithms to classify out data check for accuracy

# Algorithm - Naive Bayes Classifier
# fit the training dataset on the classifier
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_NB = Naive.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score -> ",accuracy_score(predictions_NB, Test_Y)*100)


# Algorithm - SVM Classifier
# fit the training dataset on the classifier
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_SVM = SVM.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'\ufeff.', '.'}
0    ['ذكرتني', 'بنكته', 'اللي', 'طلع', 'مع', 'بدو...
1    ['ذكرتيني', 'كمان', 'نقول', 'اذا', 'ما', 'علقت...
2    ['ذكريات', 'الباث', 'انا', 'كل', 'م', 'نويت', ...
3    ['رابطة', 'المتكممين', 'مسويه', 'التكميم', 'من...
4    ['راجل', 'طول', 'وعرض', 'وشنب', 'ويقول', 'كلام ...
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  72.89953521630318
SVM Accuracy Score ->  72.32749374329639

Process finished with exit code 0
```

II. Results of mix dialects dataset (no stemming, with Stopwords removal)

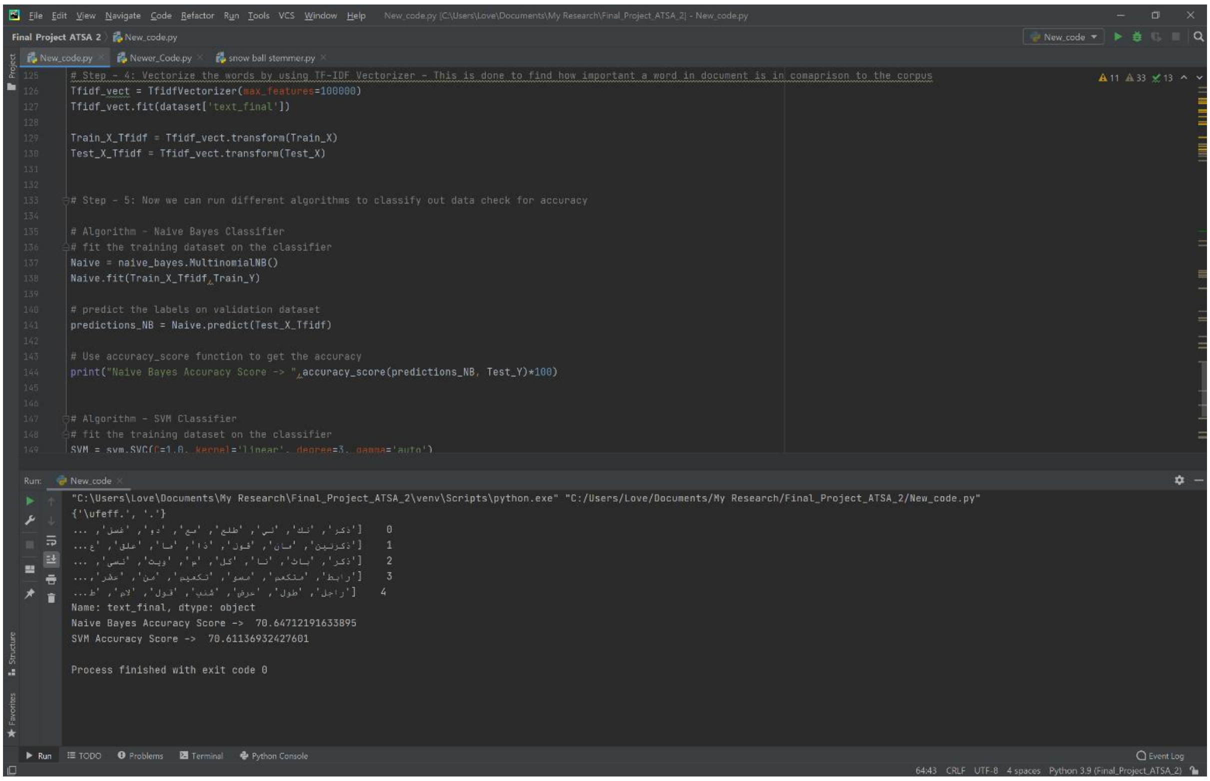


File Edit View Navigate Code Refactor Run Tools VCS Window Help
New_code.py [C:\Users\Love\Documents\My Research\Final_Project_ATSA_2] - New_code.py
Final Project ATSA 2 New_code.py
New_code.py Newer_Code.py snow ball stemmer.py
for index,entry in enumerate(da...
Run: New_code
Run TODO Problems Terminal Python Console Event Log
94:60 CRLF UTF-8 4 spaces Python 3.9 (Final_Project_ATSA_2)

```
removing emoji
'''

'''
Preprocessing the Tweet Text
'''
# Step 1: Remove blank rows if any
dataset['text'].dropna(inplace=True)

# Step 2: Tokenization : In this each entry in the corpus will be broken into set of words
dataset['text']= [word_tokenize(entry) for entry in dataset['text']]
# dataset['text']=dataset['text'].apply(lambda x: stemming(x))
# WordNetLemmatizer requires Pos tags to understand if the word is noun or verb or adjective etc. By default it is set to Noun
tag_map = defaultdict(lambda : wn.NOUN)
tag_map['J'] = wn.ADJ
tag_map['V'] = wn.VERB
tag_map['R'] = wn.ADV

for index,entry in enumerate(dataset['text']):
    # Declaring Empty List to store the words that follow the rules for this step
    Final_words = []
    # Initializing WordNetLemmatizer()
    word_Lemmatized = WordNetLemmatizer()
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'طرا', 'فانه', 'بها', 'إذ', 'إياه', 'يمكن', 'خلف', 'ج', 'كلا', 'اذا', 'هؤلاء', 'خامس', 'تسعة', 'أما', 'اكد', 'ومع', 'صه', 'آه', 'اللي', 'بن', 'ذهب', 'وجد', 'عدا', 'فبراير', 'جوان', 'عدة', 'ير
0    ['ذكرتني', 'بنكته', 'طلع', 'بدو', 'وغسل', 'الح...
1    ['ذكرتيني', 'كمان', 'نقول', 'علقتي', 'سنابالي...
2    ['ذكريات', 'الباث', 'نويت', 'انسى', 'لك', 'الذ...
3    ['رابطة', 'المتكممين', 'مسويه', 'التكميم', 'فه...
4    ['راجل', 'طول', 'وعرض', 'وشنب', 'ويقول', 'كلام ...
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  72.04147300679298
SVM Accuracy Score ->  71.96996782266714

Process finished with exit code 0
```

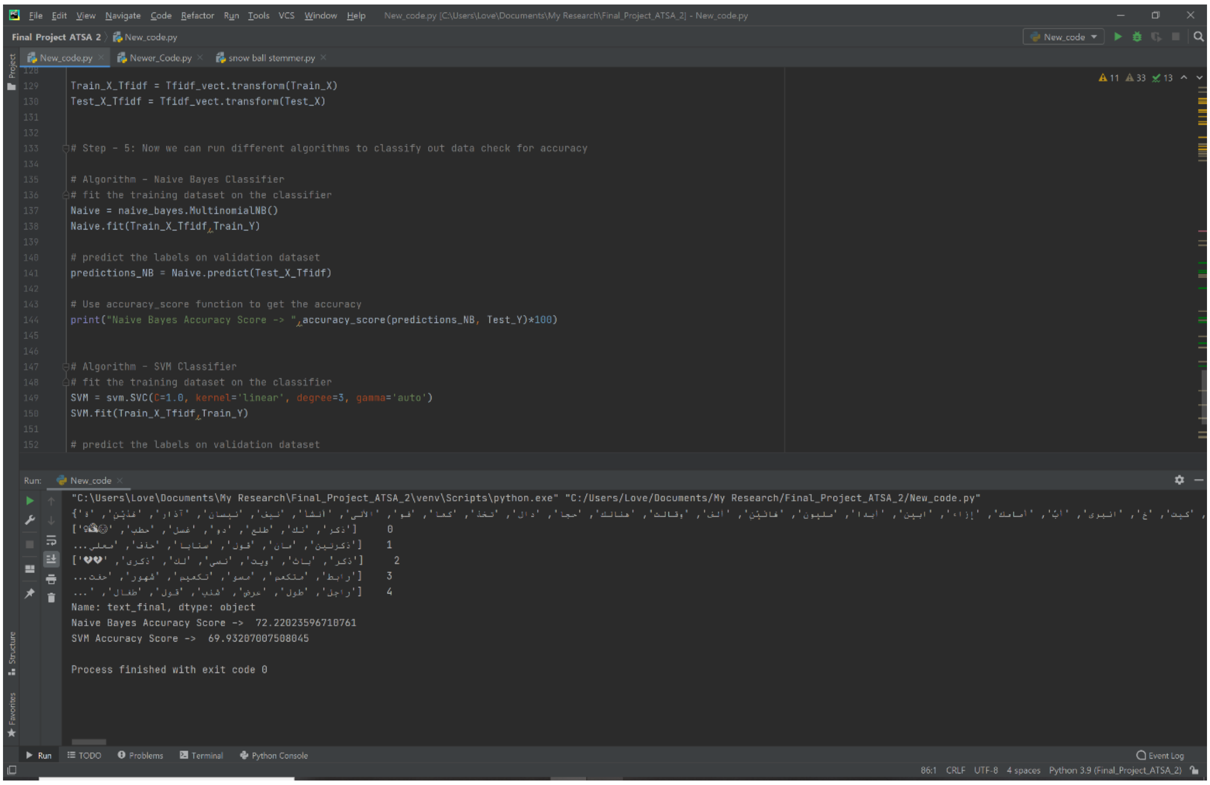
III. Results of mix dialects dataset (with stemming, no Stopwords removal)



```
# Step - 4: Vectorize the words by using TF-IDF Vectorizer - This is done to find how important a word in document is in comaprison to the corpus
Tfidf_vect = TfidfVectorizer(max_features=100000)
Tfidf_vect.fit(dataset['text_final'])

Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)


# Step - 5: Now we can run different algorithms to classify out data check for accuracy

# Algorithm - Naive Bayes Classifier
# fit the training dataset on the classifier
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_NB = Naive.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score -> ",accuracy_score(predictions_NB, Test_Y)*100)


# Algorithm - SVM Classifier
# fit the training dataset on the classifier
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'\ufeff.', '.'}
0    ['ذكر', 'نك', 'لي', 'طلع', 'مع', 'دو', 'غسل', ...
1    ['ذكرتين', 'مان', 'فول', 'ذا', 'ما', 'علق', 'ع...
2    ['ذكر', 'بات', 'نا', 'كل', 'م', 'ويت', 'نسي', ...
3    ['رابط', 'متكمم', 'مسو', 'تكميم', 'من', 'عشر',...
4    ['راجل', 'طول', 'عرض', 'شنب', 'قول', 'لام', 'ط...
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  70.64712191633895
SVM Accuracy Score ->  70.61136932427601

Process finished with exit code 0
```

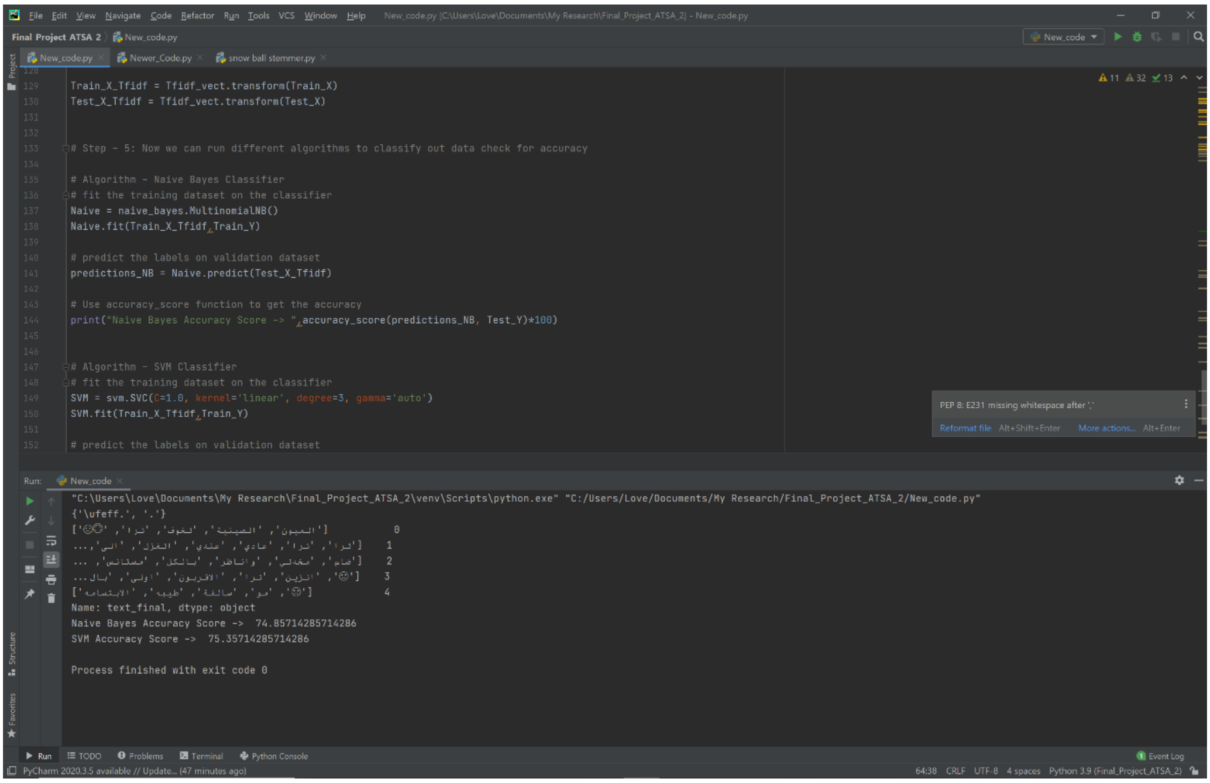
IV. Results of mix dialects dataset (with stemming, with Stopwords removal)



```
Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)


# Step - 5: Now we can run different algorithms to classify out data check for accuracy

# Algorithm - Naive Bayes Classifier
# fit the training dataset on the classifier
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_NB = Naive.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score -> ",accuracy_score(predictions_NB, Test_Y)*100)


# Algorithm - SVM Classifier
# fit the training dataset on the classifier
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'ة', 'فلذين', 'آذار', 'نيسان', 'نيف', 'انشا', 'الألى', 'فو', 'كما', 'تخذ', 'دال', 'حجا', 'هنالك', 'وقالت', 'ألف', 'فانين', 'مليون', 'أبدا', 'ابين', 'إزاء', 'أصامك', 'أبّ', 'انبرى', 'خ', 'كيت', ,
0    ['ذكر', 'نك', 'طلع', 'دو', 'غسل', 'حطب', '[illegible]']
1    ['ذكرتين', 'مان', 'فول', 'ستايا', 'حذف', 'معلي...
2    ['ذكر', 'بات', 'ويت', 'نسي', 'لك', 'ذكرى', '💔💔']
3    ['رابط', 'متكمم', 'مسو', 'تكميم', 'شهور', 'حفت...
4    ['راجل', 'طول', 'عرض', 'شنب', 'قول', 'طفال', '...
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  72.22023596710761
SVM Accuracy Score ->  69.93207007508045

Process finished with exit code 0
```

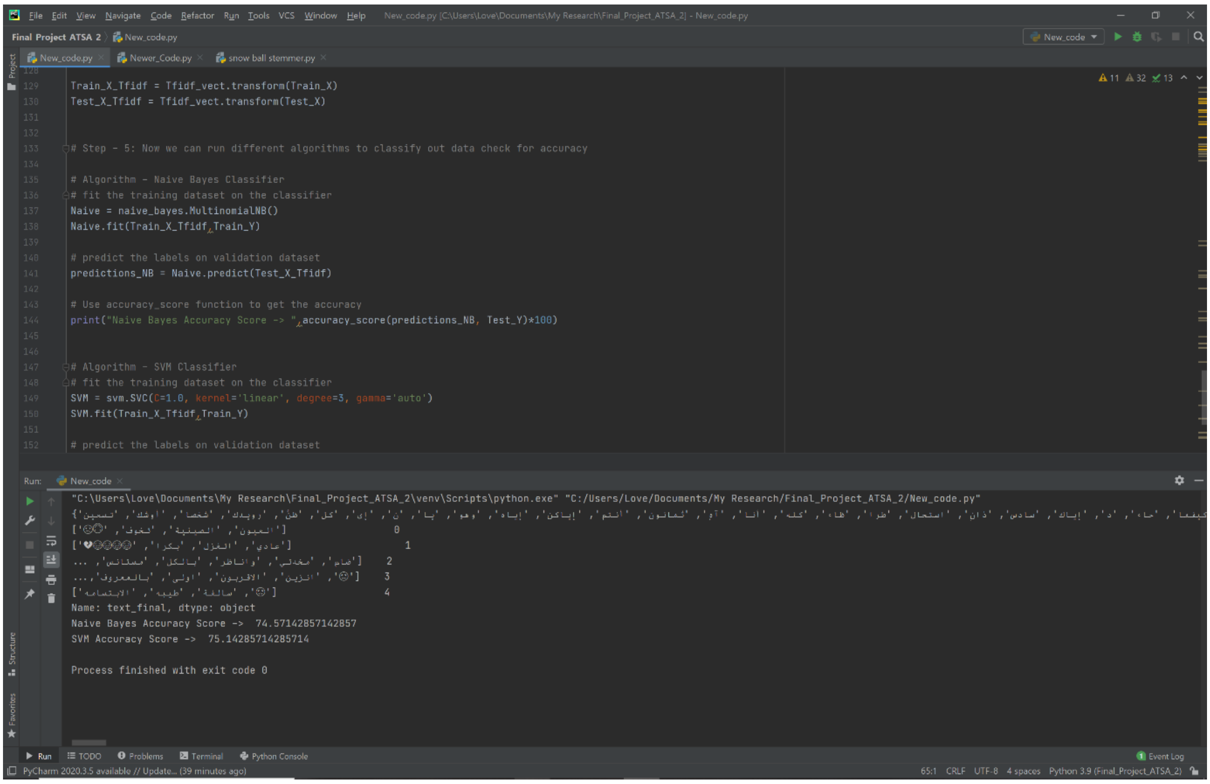
V. Results of gulf dialect dataset (no stemming, no Stopwords removal)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
New_code.py [C:\Users\Love\Documents\My Research\Final_Project_ATSA_2] - New_code.py
Final Project ATSA 2
New_code.py
Newer_Code.py
snow ball stemmer.py
New_code
PEP 8: E231 missing whitespace after ','
Reformat file Alt+Shift+Enter
More actions... Alt+Enter
Run
TODO
Problems
Terminal
Python Console
Event Log
PyCharm 2020.3.5 available // Update... (47 minutes ago)
64:38 CRLF UTF-8 4 spaces Python 3.9 (Final_Project_ATSA_2)

```
Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)


# Step - 5: Now we can run different algorithms to classify out data check for accuracy

# Algorithm - Naive Bayes Classifier
# fit the training dataset on the classifier
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_NB = Naive.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score -> ",accuracy_score(predictions_NB, Test_Y)*100)


# Algorithm - SVM Classifier
# fit the training dataset on the classifier
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'\ufeff.', '.'}
['😕😶' ,'ترا' ,'تخوف' ,'الصينية' ,'العيون']          0
...,'الى' ,'الغزل' ,'عندي' ,'عادي' ,'ترا' ,'ترا']    1
... ,'مستانس' ,'بالكل' ,'واناظر' ,'مخدني' ,'ضام']    2
...بال' ,'اولى' ,'الاقربون' ,'ترا' ,'انزين' ,'😕']    3
['الابتسامه' ,'طيبه' ,'سالفة' ,'مو' ,'😕']           4
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  74.85714285714286
SVM Accuracy Score ->  75.35714285714286

Process finished with exit code 0
```

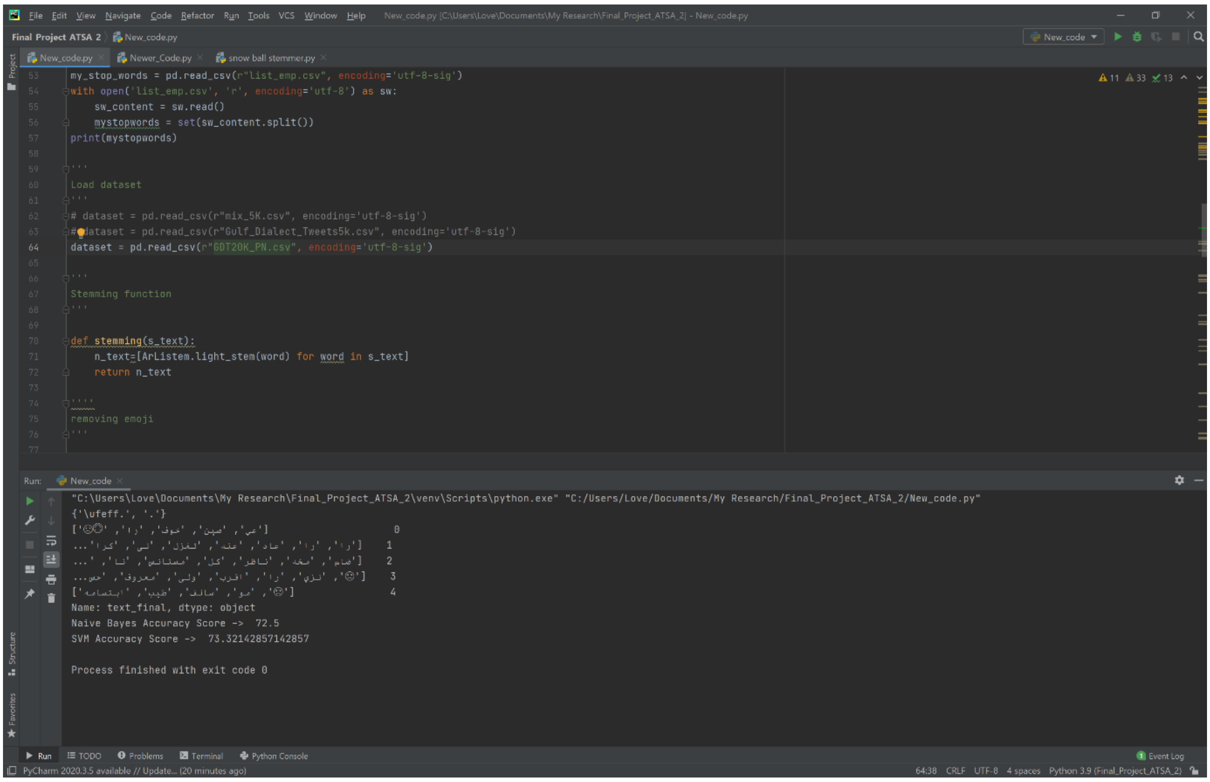
VI. Results of gulf dialect dataset (no stemming, with Stopwords removal)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
New_code.py [C:\Users\Love\Documents\My Research\Final_Project_ATSA_2] - New_code.py
Final Project ATSA 2
New_code.py
Newer_Code.py
snow ball stemmer.py
New_code
Run
TODO
Problems
Terminal
Python Console
Event Log
PyCharm 2020.3.5 available // Update... (39 minutes ago)
65:1 CRLF UTF-8 4 spaces Python 3.9 (Final_Project_ATSA_2)

```
Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)


# Step - 5: Now we can run different algorithms to classify out data check for accuracy

# Algorithm - Naive Bayes Classifier
# fit the training dataset on the classifier
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_NB = Naive.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score -> ",accuracy_score(predictions_NB, Test_Y)*100)


# Algorithm - SVM Classifier
# fit the training dataset on the classifier
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'تسعين' ,'أوشك' ,'شخصا' ,'رويدك' ,'ظنّ' ,'كل' ,'إي' ,'ن' ,'يا' ,'وهو' ,'إياه' ,'إياكن' ,'أنتم' ,'ثمانون' ,'آه' ,'أنا' ,'كله' ,'هاء' ,'طرا' ,'استحال' ,'ذان' ,'سادس' ,'إياك' ,'د' ,'حاء' ,'كيفما
['😕😶' ,'تخوف' ,'الصينية' ,'العيون']                  0
['💔😩😩😩😩' ,'بكرا' ,'الغزل' ,'عادي']                    1
... ,'مستانس' ,'بالكل' ,'واناظر' ,'مخدني' ,'ضام']    2
...,'بالمعروف' ,'اولى' ,'الاقربون' ,'انزين' ,'😕']    3
['الابتسامه' ,'طيبه' ,'سالفة' ,'😕']                  4
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  74.57142857142857
SVM Accuracy Score ->  75.14285714285714

Process finished with exit code 0
```

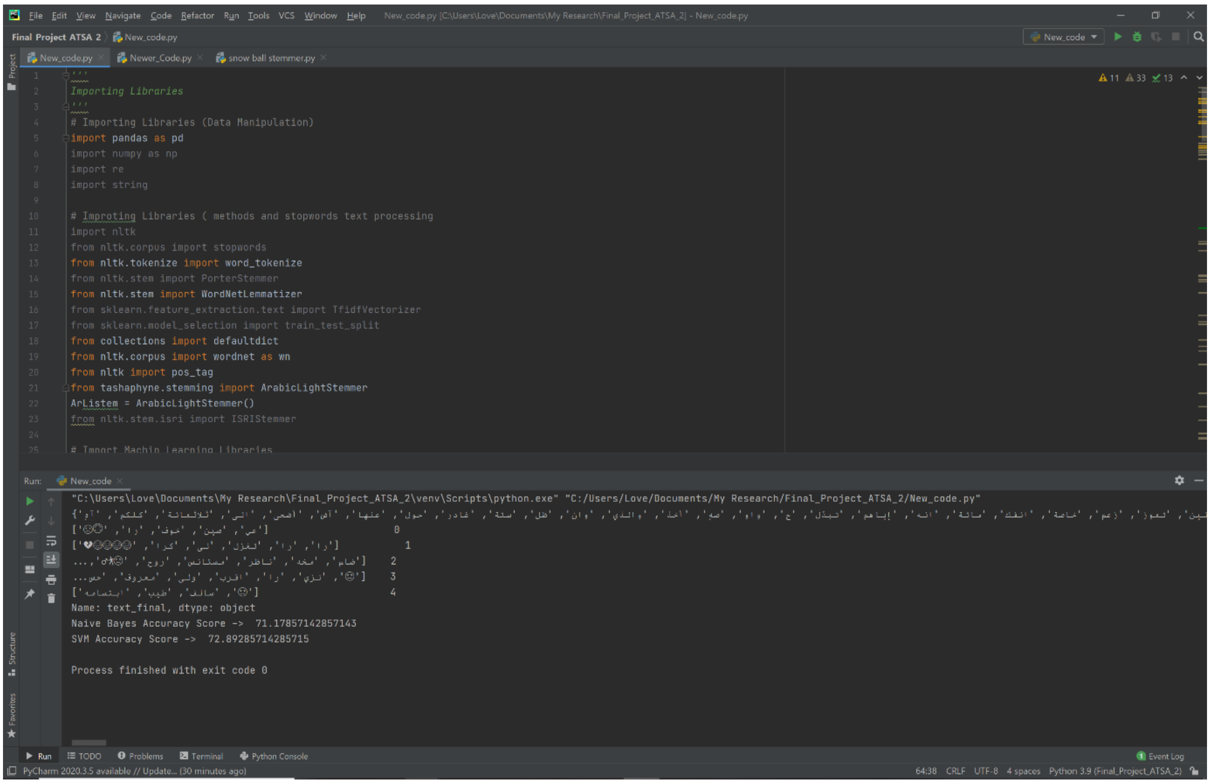
VII. Results of gulf dialect dataset (with stemming, no Stopwords removal)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
New_code.py [C:\Users\Love\Documents\My Research\Final_Project_ATSA_2] - New_code.py
Final Project ATSA 2 New_code.py
New_code.py
Newer_Code.py
snow ball stemmer.py
Run: New_code
Run
TODO
Problems
Terminal
Python Console
Event Log
PyCharm 2020.3.5 available // Update... (20 minutes ago)
64:38 CRLF UTF-8 4 spaces Python 3.9 (Final_Project_ATSA_2)

```python
my_stop_words = pd.read_csv(r"list_emp.csv", encoding='utf-8-sig')
with open('list_emp.csv', 'r', encoding='utf-8') as sw:
    sw_content = sw.read()
    mystopwords = set(sw_content.split())
print(mystopwords)

'''
Load dataset
'''
# dataset = pd.read_csv(r"mix_5K.csv", encoding='utf-8-sig')
#dataset = pd.read_csv(r"Gulf_Dialect_Tweets5k.csv", encoding='utf-8-sig')
dataset = pd.read_csv(r"GDT20K_PN.csv", encoding='utf-8-sig')

'''
Stemming function
'''

def stemming(s_text):
    n_text=[ArListem.light_stem(word) for word in s_text]
    return n_text

''''
removing emoji
'''
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'\ufeff.', '.'}
['😊😊' ,'ر ا' ,'خوف' ,'صبن' ,'عي']                      0
...'كرا' ,'لى' ,'لغزل' ,'عنه' ,'عاد' ,'ر ا' ,'ر ا']     1
...' ,'نا' ,'مستانس' ,'كل' ,'ناظر' ,'مخه' ,'ضام']     2
...حس' ,'معروف' ,'ولي' ,'اقرب' ,'را' ,'نزي' ,'😊']    3
['ابتسامه' ,'طيب' ,'سالف' ,'مو' ,'😊']                  4
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  72.5
SVM Accuracy Score ->  73.32142857142857

Process finished with exit code 0
```

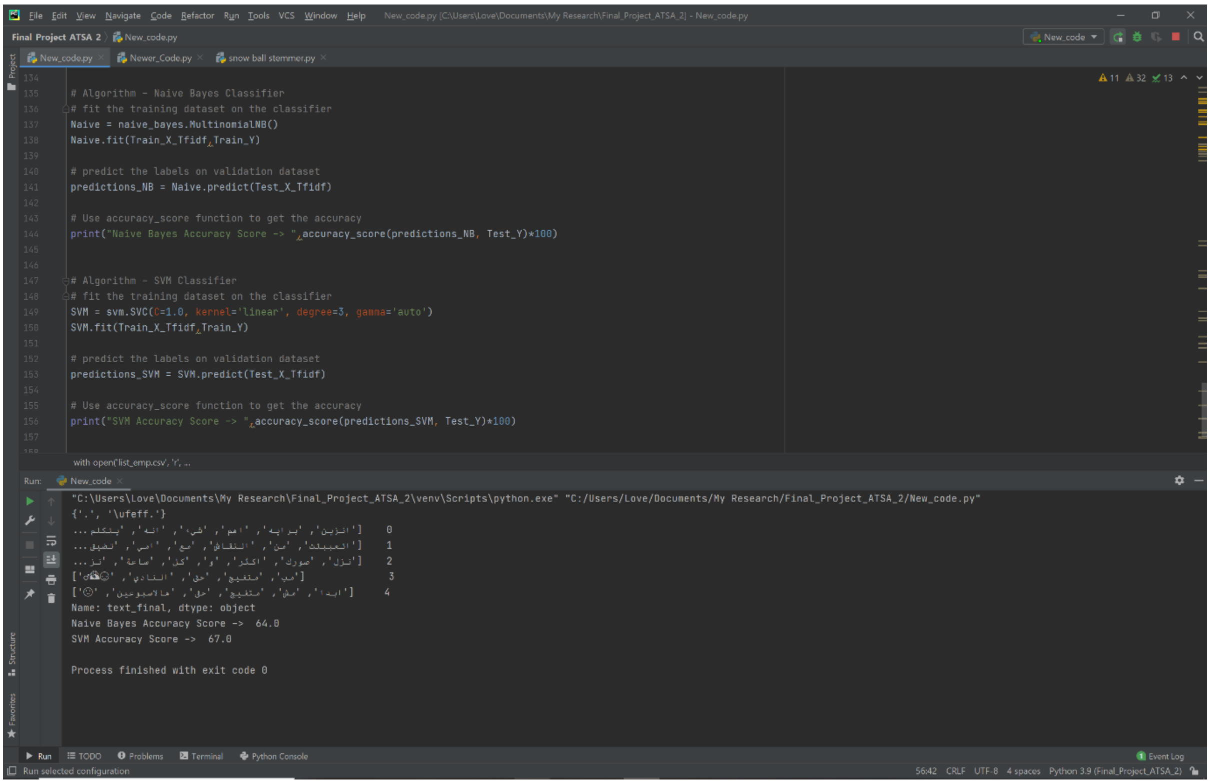
VIII. Results of gulf dialect dataset (with stemming, with Stopwords removal)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
New_code.py [C:\Users\Love\Documents\My Research\Final_Project_ATSA_2] - New_code.py
Final Project ATSA 2 New_code.py
New_code.py
Newer_Code.py
snow ball stemmer.py
Run: New_code
Run
TODO
Problems
Terminal
Python Console
Event Log
PyCharm 2020.3.5 available // Update... (30 minutes ago)
64:38 CRLF UTF-8 4 spaces Python 3.9 (Final_Project_ATSA_2)

```python
'''
Importing Libraries
'''
# Importing Libraries (Data Manipulation)
import pandas as pd
import numpy as np
import re
import string

# Improting Libraries ( methods and stopwords text processing
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from collections import defaultdict
from nltk.corpus import wordnet as wn
from nltk import pos_tag
from tashaphyne.stemming import ArabicLightStemmer
ArListem = ArabicLightStemmer()
from nltk.stem.isri import ISRIStemmer

# Import Machin Learning Libraries
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'آ و' ,'كلكم' ,'ثلاثمائة' ,'الى' ,'اضحى' ,'آض' ,'عنها' ,'حول' ,'غادر' ,'ستة' ,'ظل' ,'وان' ,'والذي' ,'اخذ' ,'صبر' ,'واو' ,'ج' ,'تبدل' ,'إياهم' ,'انه' ,'مائة' ,'انفك' ,'خاصة' ,'زعم' ,'تعوذ' ,'بين
['😊😊' ,'ر ا' ,'خوف' ,'صبن' ,'عي']                      0
['♥😊😊😊😊' ,'كرا' ,'لى' ,'لغزل' ,'را' ,'ر ا']            1
...,'♂☆😊' ,'روج' ,'مستانس' ,'ناظر' ,'مخه' ,'ضام']    2
...حس' ,'معروف' ,'ولي' ,'اقرب' ,'را' ,'نزي' ,'😊']    3
['ابتسامه' ,'طيب' ,'سالف' ,'😊']                        4
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  71.17857142857143
SVM Accuracy Score ->  72.89285714285715

Process finished with exit code 0
```

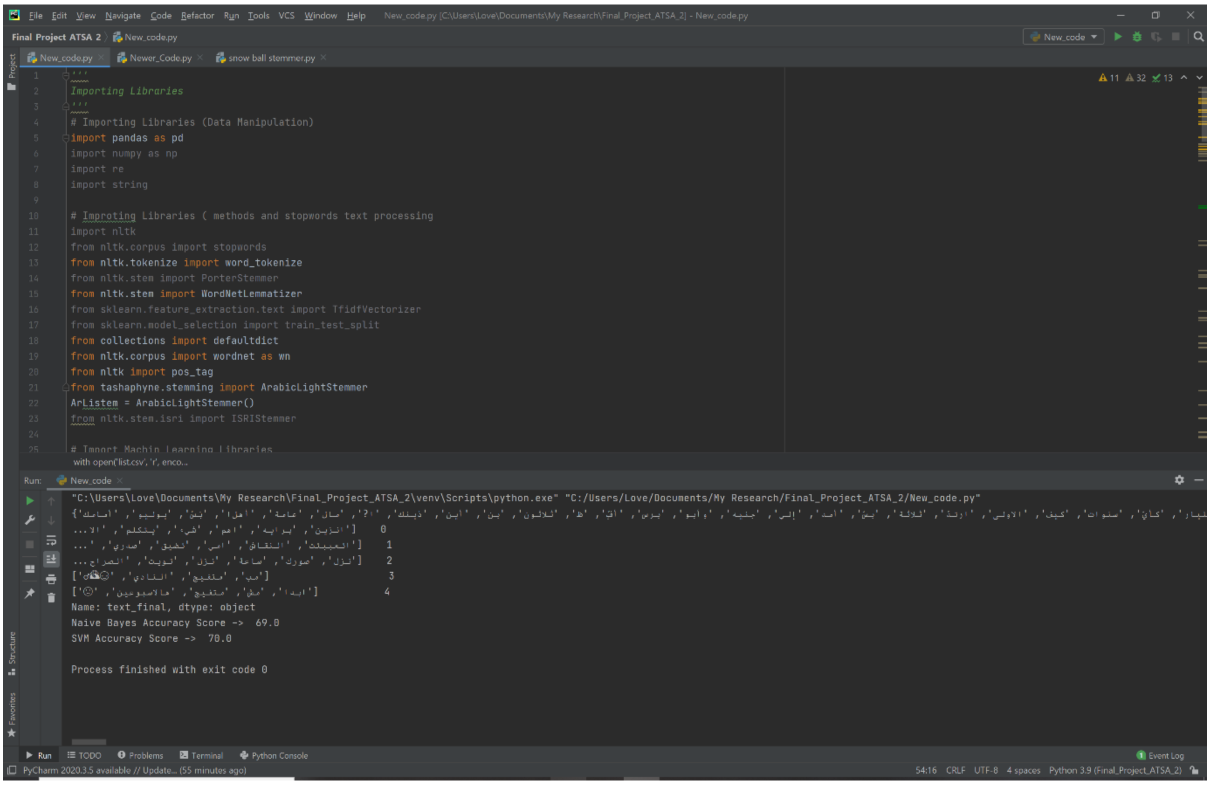
IX. Results of Emirati dialect dataset (no stemming, no Stopwords removal)



```
# Algorithm - Naive Bayes Classifier
# fit the training dataset on the classifier
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_NB = Naive.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score -> ",accuracy_score(predictions_NB, Test_Y)*100)


# Algorithm - SVM Classifier
# fit the training dataset on the classifier
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_SVM = SVM.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("SVM Accuracy Score -> ",accuracy_score(predictions_SVM, Test_Y)*100)
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'.', '\ufeff.'}
['انزين', 'برايه', 'اهم', 'شيء', 'انه', 'يتكلم ...    0
['العبيلت', 'من', 'النقاش', 'مع', 'امي', 'تضيق ...    1
['نزل', 'صورك', 'اكثر', 'و', 'كل', 'ساعة', 'نز ...    2
['مب', 'متغيج', 'حق', 'النادي', '♂🏕☹']                3
['ابدا', 'مش', 'متغيج', 'حل', 'هالاسبوعين', '☺']      4
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  64.0
SVM Accuracy Score ->  67.0

Process finished with exit code 0
```

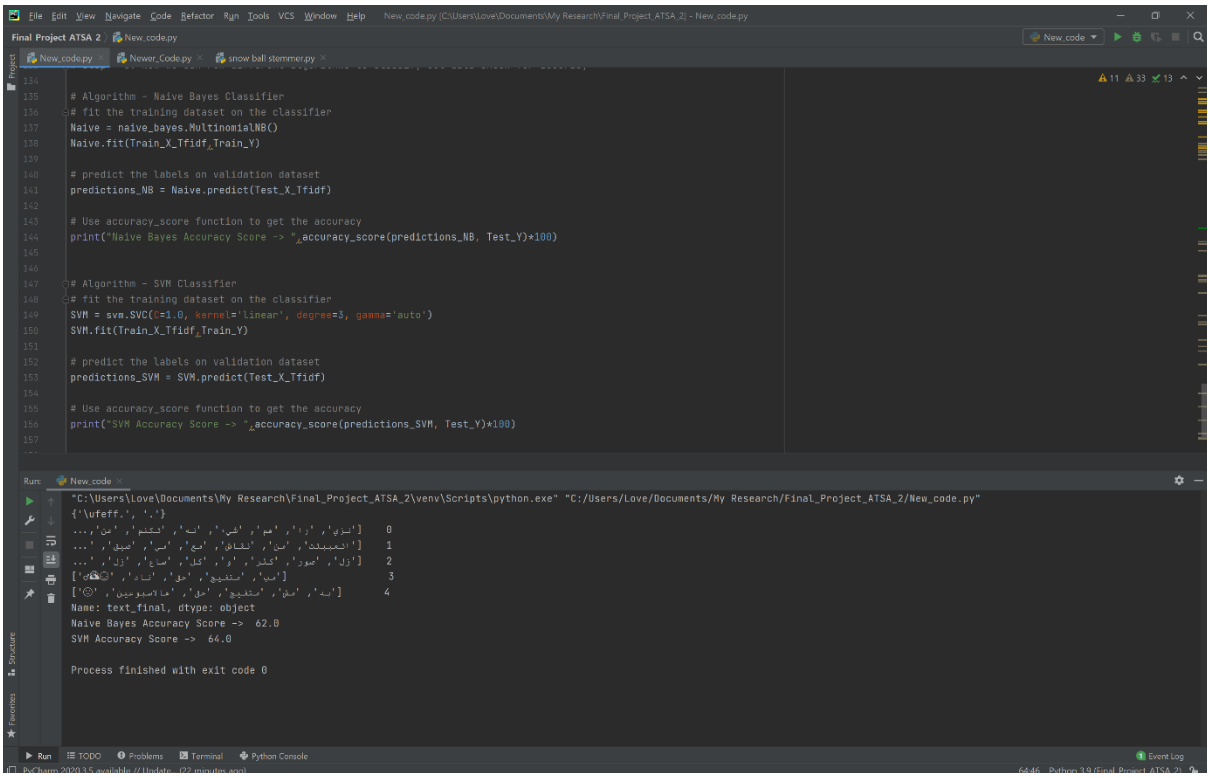
X. Results of Emirati dialect dataset (no stemming, with Stopwords removal)



```
'''
Importing Libraries
'''
# Importing Libraries (Data Manipulation)
import pandas as pd
import numpy as np
import re
import string

# Improting Libraries ( methods and stopwords text processing
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from collections import defaultdict
from nltk.corpus import wordnet as wn
from nltk import pos_tag
from tashaphyne.stemming import ArabicLightStemmer
ArListem = ArabicLightStemmer()
from nltk.stem.isri import ISRIStemmer

# Import Machin Learning Libraries
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'اماملك', 'يوليو', 'بتن', 'أهلا', 'عامة', 'مال', '?', 'ذينك', 'ابن', 'بن', 'ثلاثون', 'ظ', 'اق', 'برس', 'وابو', 'جنيه', 'إلى', 'أمد', 'بما', 'ثلاثة', 'ارتد', 'الاولى', 'كيف', 'سنوات', 'كاي', 'ليار
['انزين', 'برايه', 'اهم', 'شيء', 'يتكلم', 'الا ...    0
['العبيلت', 'النقاش', 'امي', 'تضيق', 'صدري', ' ...    1
['نزل', 'صورك', 'ساعة', 'نزل', 'تويت', 'الصراح ...    2
['مب', 'متغيج', 'النادي', '♂🏕☹']                      3
['ابدا', 'مش', 'متغيج', 'هالاسبوعين', '☺']            4
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  69.0
SVM Accuracy Score ->  70.0

Process finished with exit code 0
```

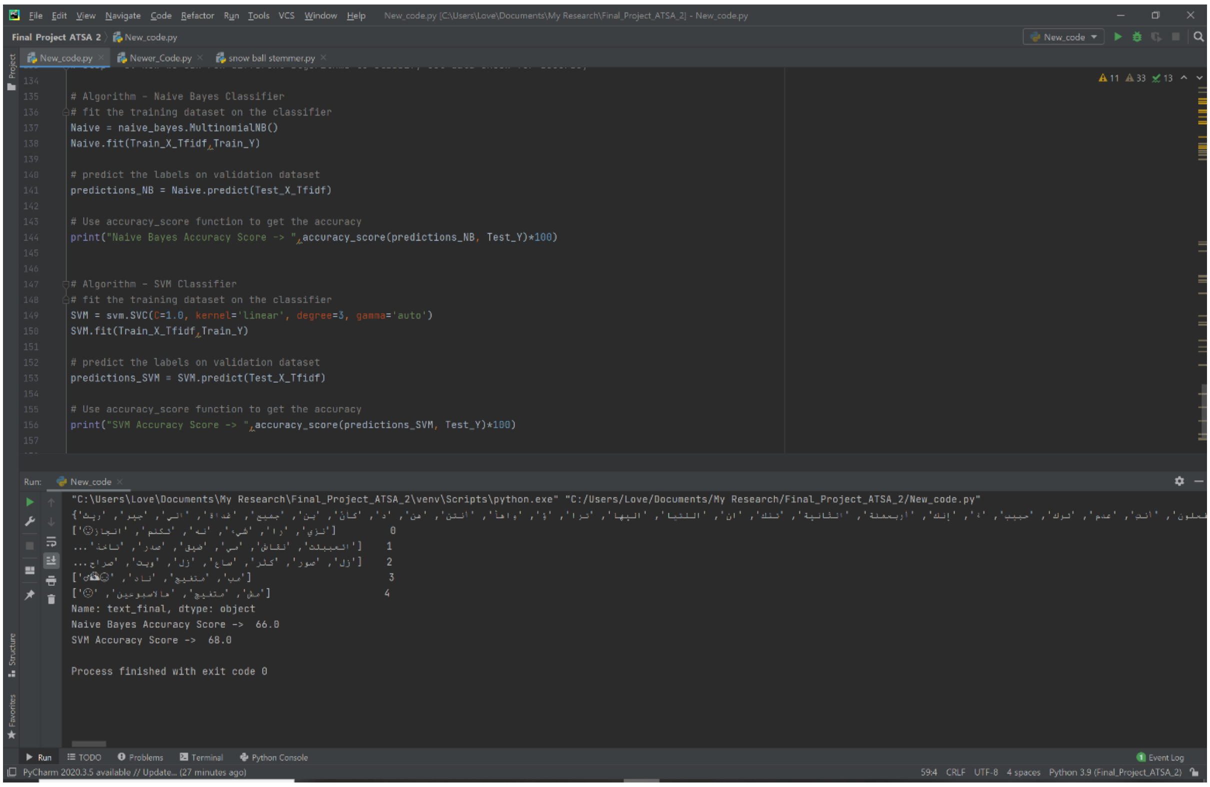
XI. Results of Emirati dialect dataset (with stemming, no Stopwords removal)



```
# Algorithm - Naive Bayes Classifier
# fit the training dataset on the classifier
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_NB = Naive.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score -> ",accuracy_score(predictions_NB, Test_Y)*100)


# Algorithm - SVM Classifier
# fit the training dataset on the classifier
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_SVM = SVM.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("SVM Accuracy Score -> ",accuracy_score(predictions_SVM, Test_Y)*100)
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'\ufeff.', '.'}
0    ['نزي', 'را', 'هم', 'شيء', 'نه', 'تكلم', 'عن',...
1    ['العبيلت', 'من', 'نقاش', 'مع', 'مي', 'ضيق', '...
2    ['زل', 'صور', 'كثر', 'و', 'كل', 'ساع', 'زل', '...
3                 ['مب', 'متفيج', 'حق', 'ناد', '♂🐸☺']
4      ['بد', 'مش', 'متفيج', 'حق', 'هالاسبوعين', '☺']
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  62.0
SVM Accuracy Score ->  64.0

Process finished with exit code 0
```

XII. Results of Emirati dialect dataset (with stemming, with Stopwords removal)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Final Project ATSA 2
New_code.py
Newer_Code.py
snow ball stemmer.py
Run:
New_code
Run
TODO
Problems
Terminal
Python Console
Event Log
PyCharm 2020.3.5 available // Update... (27 minutes ago)
59:4 CRLF UTF-8 4 spaces Python 3.9 (Final_Project_ATSA_2)

```
# Algorithm - Naive Bayes Classifier
# fit the training dataset on the classifier
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_NB = Naive.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("Naive Bayes Accuracy Score -> ",accuracy_score(predictions_NB, Test_Y)*100)


# Algorithm - SVM Classifier
# fit the training dataset on the classifier
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)

# predict the labels on validation dataset
predictions_SVM = SVM.predict(Test_X_Tfidf)

# Use accuracy_score function to get the accuracy
print("SVM Accuracy Score -> ",accuracy_score(predictions_SVM, Test_Y)*100)
```

```
"C:\Users\Love\Documents\My Research\Final_Project_ATSA_2\venv\Scripts\python.exe" "C:/Users/Love/Documents/My Research/Final_Project_ATSA_2/New_code.py"
{'ريت', 'جير', 'الي', 'غداة', 'جميع', 'ين', 'كان', 'د', 'هن', 'انتن', 'واها', 'ؤ', 'ترا', 'اليها', 'اللتيا', 'ان', 'ذلك', 'الثانية', 'اربعمئة', 'إنك', 'ذ', 'حبيب', 'ترك', 'عدم', 'انتم', 'فعلون
0         ['نزي', 'را', 'شيء', 'نه', 'تكلم', 'انجاز☺']
1    ['العبيلت', 'نقاش', 'مي', 'ضيق', 'صدر', 'ناخة'...
2    ['زل', 'صور', 'كثر', 'ساع', 'زل', 'ويت', 'صراح...
3                       ['مب', 'متفيج', 'ناد', '♂🐸☺']
4                 ['مش', 'متفيج', 'هالاسبوعين', '☺']
Name: text_final, dtype: object
Naive Bayes Accuracy Score ->  66.0
SVM Accuracy Score ->  68.0

Process finished with exit code 0
```