IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

SVM Based Approach for Detecting Sarcasm in Arabic Text

Mohammed M. Abuteir[1], Eltyeb S. A. Elsamani[2]

Department of Information Technology, College of Graduate Studies and Scientific Research,

University of Al-Buatana, Sudan[1]

University College of Science and Technology - Khan Yunis, Palestine[1]

Department of Computer Science, Faculty of Computer Science and Information Technology,
Al-Neelain University, Sudan[2]

Abstract: Sentiment Analysis (SA) is the process of determining the opinion of a text written in a natural language to be positive, negative, or neutral towards any specific target such as individuals, events, topics, products, organizations, services, etc. SA has its challenges, and one of them is sarcasm. Sarcasm is a form of communication that is intended to mock or harass someone by using words with the opposite of their literal meaning. It is often used to express a negative message using positive words. However, Sarcasm detection is one of the most challenging tasks in Natural Language Processing (NLP) especially for the Arabic language which has a rich nature and very complex morphology. It has gained relevance recently, due to its importance in improving the performance of various NLP applications including SA. In this paper, we propose an approach for automatic sarcasm detection in the Arabic text of Twitter data by using the Support Vector Machine (SVM) classifier to classify sarcastic tweets based on different N-gram features and using several weighting schemes. The experimental results obtained are promising. The best results by SVM classifier for all feature sets and several weighting schemes achieved overall accuracy equal to 86.60%, which these results are quite high especially regarding Arabic text.

Keywords: Automatic sarcasm detection; Sarcasm; Sentiment analysis; Text Mining; Support Vector Machine (SVM); Arabic language

I. INTRODUCTION

Social media or social networks can be defined as a group of internet-based applications that build on the technological foundations of Web 2.0 that allows the creation and exchange of user-generated contents [1]. Social media has continued to evolve and has become an integral part of people's daily lives, and hence play a role in modern society. Through social media networks we can share news, exchange information and express opinions and ideas without difficulty [2]. It has become a significant source of data in a variety of scientific and humanistic fields, and provide a rich vein of information that is easily mined [3]. Most famous of these social networks are Facebook and Twitter, as of the first quarter of 2021 and according to their official websites [4, 5], Facebook and Twitter have more than 2.8 billion and 353 million monthly active users respectively. Twitter has become a melting pot for all-ordinary individuals, celebrities, politicians, companies, activists, etc. It has rapidly established as an emerging global communication service due to its timeliness, convenience as a lightweight and easy way of communication and information sharing. Beyond merely displaying news and reports, the Twitter itself is also a large platform where different opinions are presented and exchanged [6]. These opinions are expressed in various forms such as articles, reviews, forum posts, short comments, tweets, etc. [7]. Opinions are very important. Whenever we need to make a decision we need to know others opinion. This is not only true for individuals, but it is also true for organizations and governments [8]. These trends are opening up the era of SA. The main purpose of SA is to identify the semantic orientation (positive, negative, or neutral) in a given piece of text. This has many benefits for businesses, education, commerce, health and many more. Despite the high usage of the SA applications, there are still rooms for improvements. One of the problems that still become a challenge in SA is sarcasm.

Sarcasm is a form of communication that is intended to mock or harass someone by using words with the opposite of their literal meaning. It is often used to express a negative message using positive words as shown in the following examples taken from our experimental corpus.

- عن جد هالحكومة عاطيتنا حريتنا بالكامل هههههههه
  (This government really gives us full freedom hahahaha)
- بجد شفيق ده متعة ...أهو ينفع بدل عكشة..ياريت يستضيفوه كتير الفترة اللي جاية
  (Seriously, Shafiq, this is a pleasure ... Is he useful instead of Okasha..I wish they would host it for a long time)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

- مبروك علينا العراق الجديد ....
  (Congratulations to us, the new Iraq ....)
- أشياء تجلب السعادة يصير عندك صديق نصراوي !!!!
  (Things that bring happiness. You have a friend, Nasrawi !!!!)
- النصر باقي له نقطه ويضمن البقاء بالدوري السعودي كبير ياعالمي .....
  (Al-Nasr has a point remaining and guarantees staying in the Saudi League, a great world .....)

An exact, universal definition of sarcasm is hard to nail down. The online Oxford dictionary [9] defines sarcasm as: "The use of irony to make or convey contempt". Merriam Webster [10] gives several definitions:

- *"A sharp and often satirical or ironic utterance designed to cut or give pain".*
- *"A mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual".*
- *"The use of words that mean the opposite of what you really want to say especially in order to insult someone, to show irritation".*

The last definition is closest to the one that will be used here; for the purpose of this paper, sarcasm will be defined as follows: "Sarcasm is meaning the opposite of what you say". This definition was chosen because it is well defined and allows for relatively precise classification of tweets as sarcastic or not. The detection of sarcasm is important, if not crucial, for the development and refinement of SA systems and it will results in higher classifier accuracy, but it is at the same time a serious conceptual and technical challenge [11]. Sarcasm is an old and well researched phenomenon in the field of linguistic psychology and cognitive science [12]. Unfortunately, in the field of text mining or more specifically: SA, detecting sarcasm automatically is still considered a challenging task. Automatic sarcasm detection refers to computational approaches to predict if a given text is sarcastic or not. This is a crucial step to SA, considering prevalence and challenges of sarcasm in sentiment-bearing text especially for the Arabic language which has a rich nature and very complex morphology. More formally, sarcasm detection on Twitter can be defined as follows:

Given an unlabeled tweet t from user U along with a set of U's past tweets T, a solution to sarcasm detection aims to automatically detect if t is sarcastic or not.

The issue of automatic sarcasm detection has been addressed mostly in English, although there has been some research in other languages. The Arabic language is challenging for many Natural Language Processing NLP tasks because of their rich morphology and syntax. This has motivated us to focus our current research on the Arabic language to establish the state of the art baselines for sarcasm detection in the Arabic language.

The Arabic Language is one of the most widely used languages in the world. It is spoken by more than 422 million speakers in the world [13]. It is not like European languages, such as English, because of its richer morphological structure. It also has many challenges that require special processing. Therefore, Arabic NLP has become attractive to researchers due to its complexity and the scarcity of available resources; as a result, the importance of addressing this language has been noted. It can be seen that strong effort is being made with the fundamental tools of NLP in Arabic, such as the morphological analyzer, part of speech tagger, and syntactical parser. The field of Arabic NLP is still at an early stage of evolution. Nevertheless, work in some areas, such as text classification, sentiment analysis, is beginning to appear [14].

In this paper, we propose an approach for automatic detection of sarcasm in the Arabic tweets. Our approach is based on using the Support Vector Machine (SVM) classifier to classify sarcastic tweets based on different N-gram features and using several weighting schemes. These features are extraced from the corpus that we have collected from Twitter. Our corpus consists of a total of 20000 tweets (10000 are sarcastic tweets and 10000 are non sarcastic).

The rest of this paper is organized as follows: Section II presents the related works. Section III describes our proposed approach for sarcasm detection in the Arabic tweets. Section IV presents our experiments and discusses the obtained results. Finally, Section V presents the conclusion and future work.

II. RELATED WORKS

In the text mining literature, automatic detection of sarcasm is considered a challenging task. It has become a researched subject in recent years due to its practical implications in social media platforms. The majority of this research has been done in English, as this is the dominant language of science. Recently, a few researchers have concentrated on sarcasm detection to other languages.

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

Sarcasm detection has been modeled as a binary classification problem, where mostly tweets labeled with certain hashtags (i.e., #sarcasm, #sarcastic) have been considered as sarcastic utterances. Following this framework, different approaches in different languages have been proposed.

Carvalho et al. [15], created an automatic system for detecting irony. They investigated the use of a set of pre-defined surface patterns (i.e., emoticons, laughter expressions, heavy punctuation marks, quotation marks, and positive interjections). They focused on detection of ironic style in comments on articles from a Portuguese online newspaper.The initial focus is on identifying irony in sentences containing positive predicates since these sentences are more exposed to irony, making their true polarity harder to recognize. Their collection is composed of 8,211 news and corresponding comments posted by on-line readers. It includes about 250,000 user posts, totaling approximately one million sentences. They showed that it is possible to find ironic sentences with relatively high precision (from 45% to 85%) by exploring certain oral or gestural clues in user comments. They also demonstrated that clues based on deeper linguistic information are relatively inefficient in capturing irony in user-generated content, which points to the need for exploring additional types of oral clues.

Liebrecht et al. [16], used a Balanced Winnow classifier to detects sarcastic tweets. They collected a training corpus of about 78 thousand Dutch tweets with #sarcasm hash tag; they assuming that the human labeling is correct, annotation of a sample indicates that about 85% of these tweets are indeed sarcastic. They used unigrams, bigrams and trigrams as features and trained a machine learning classifier on the collected training corpus, and apply it to a test set of 3.3 million Dutch tweets posted on a single day. The results show that the classifier attained only a 30% average precision. The obtained precision is an indication of the difficulty of the task. These results indicate that the lexical features considered in this paper are not sufficient to accurately identify sarcastic tweets from positive and negative tweets.

Riloff et al. [17], used a well constructed lexicon-based approach to detect sarcasm in tweets based on an assumption that sarcastic tweets are a contrast between a positive sentiment and a negative situation. This approach uses a novel bootstrapping algorithm that automatically learns lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. It classifies tweets as sarcastic if it contains a positive predicative that precedes a negative situation phrase in close proximity. Their algorithm keeps iteration between two steps. The first step is learning negative situation phrases following positive sentiment, where "love" is used as an initial seed word. Then, the second step will learn positive sentiment phrases that occur near negative situation phrases. After multiple iteration processes, the obtained list of negative situations and positive sentiment phrases are used to recognize sarcasm in tweets by identifying contexts that contain a positive sentiment in close proximity (occurring nearby) to a negative situation phrase. This approach relies on the assumption that many sarcastic tweets contains the following structure: [Positive Verb Phrase][Negative Situation Phrase]. For training the algorithm to generate the lexicon, 35,000 tweets are collected with the sarcasm hashtag (#sarcasm) from Twitter. Their evaluation on a human-annotated dataset of 3000 tweets (23% sarcastic) was done using the SVM classifier with unigrams and bigrams of the learned phrases as features, achieving an F-measure of 48%. The hybrid approach that combines the results of the SVM classifier and their contrast method achieved an F-measure of 51%. The obtained F-measure is an indication of the difficulty of the task. Their approach shows some potentials. However, not all of the sarcastic tweets in Twitter fall in the aforementioned category of sarcasm. In addition, the approach relies on the existence of the all possible negative situations on the training set, which makes it less efficient when dealing with new tweets. It also cannot identify sarcasm accurately across multiple sentences.

Tungthamthiti et al. [18], proposed a method to detect sarcasm in Twitter tweets. It is based on a variety of approaches, including lexicons based SA, concept level knowledge expansion, coherence of sentences, and supervised learning classification. They used sentiment scores of words, Punctuation and special symbols as features for the classifier. They also used the common-sense concept to find the sentiment score for the word with unknown sentiment score; According to them words like raining, bad weather is conceptually same. So, if raining is present in tweet then consider it as a negative situation. Then, They consider coherence in a tweet to ensure that the tweets with contradiction in the sentiment score have dependent relationships across multiple sentences. They collected 50,000 tweets from Twitter for their corpus. 25,000 tweets were randomly selected as normal tweets, whereas the other 25,000 tweets are sarcastic tweets. Finally, They construct the feature vector to train an SVM classifier based on their proposed features and N-gram features. The results show that their method has the best accuracy when they combine their proposed features with N-gram features(unigrams, bigrams and trigrams) and it has achieved accuracy of 79.43%.

Bharti et al. [19], presented two algorithms to detect sarcasm in the text of Twitter data for two different types of tweet. The first is a parsing-based lexicon generation algorithm that generates a lexicon to identify sarcasm in tweets. Where, tweets sentiment contradicts with the tweets situation. Generated lexicon contains phrase in four categories, namely positive sentiment, negative situation, negative sentiment and positive situation, and the second is to detect sarcasm based on the occurrence of the interjection word such as oh, wow, yay, yeah, nah, aha, uh, etc. For training the first algorithm to generate the lexicon, 50000 tweets are collected with the sarcasm hashtag (#sarcasm) from Twitter with keyword love, amazing, good, hate, sad, happy, bad, hurt, awesome, excited, nice, great, sick, etc. For testing, Tweets are collected in two categories: tweets with sarcasm hashtag and tweets without hashtag. To test the first algorithm, 1500 random tweets collected with sarcasm hashtag and 1500 random tweets collected without hashtag are used. Similarly, to test the second Algorithm ,1000 tweets were collected with sarcasm hashtag and 2500 tweets without any hashtag. For

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

the second algorithm, both 1000 and 2500 tweets start with an interjection word. The first algorithm achieved 89%, 81% and 84% precision, recall and f-score respectively in tweets with sarcastic hashtag and 64%, 75% and 69% precision, recall and f-score respectively in tweets without sarcastic hashtag. The second algorithm achieved 85%, 96% and 90% precision, recall and f-score respectively in tweets with sarcastic hashtag and 77%, 73% and 74% precision, recall and f-score respectively in tweets without sarcastic hashtag. However, this approach has some limitations since not all of the sarcastic tweets in Twitter fall in the aforementioned category of sarcasm. In addition, the approach relies on the existence of the all possible positive and negative situations on the training set, which makes it less efficient when dealing with new tweets. It also cannot identify sarcasm accurately across multiple sentences.

Finally, Bouazizi and Ohtsuki [20], proposed an approach to detect sarcasm in tweets. It is based on four groups of features that covered different types of sarcasm. The features were sentiment-related features when a positive statement is collected in a negative situation; punctuation-related features for calculating the number of exclamation marks, question marks, dots marks, quotes marks, and number of capital words in each tweet; lexical and syntactic features for extracting uncommon words, common sarcastic expressions, interjections, and laughter; and patternrelated features for generating a vector of common sarcastic words with specified length. Then, machine learning algorithms were used to classify tweets into sarcastic or nonsarcastic based on three collected datasets: the first set was 6000 tweets were manually checked and used for training, the second set was 1128 tweets for optimization and the third set had 500 sarcastic tweets and 500 nonsarcastic tweets that were manually checked and used for testing. The most important set of features was pattern-related features, because it has the highest accuracy, precision, and recall results (90.0%, 90.6%, and 89.3%, respectively). Afterwards, they evaluated the classifier using all four features by cross validation and test set methods, and the results were 83.1%, 91.1%, and 73.4% for the accuracy, precision, and recall values, respectively. Also, F-measure was calculated and got 81.3%.

The previous related works confirm the growing interest for automatic sarcasm detection task in the research community, especially for understanding the impact of the sarcastic devices on SA.

III. THE PROPOSED APPROACH

This section presents our proposed approach for detecting sarcasm in the Arabic text of Twitter data. Various stages have to be performed to achieve sarcasm detection, including text data collection, text preprocessing, features extraction, SVM classification, and evaluation. The Various stages of our proposed approach is depicted in Fig. 1. First, an input tweet from collected Arabic text corpus is preprocessed by applying text cleaning, tokenization, stopwords removal, stemming, and pruning. Next, different N-gram faetures are extraced. The SVM classifier is applied to judge whether the given tweet is sarcastic or not. The SVM classifier is trained from labeled data, i.e., a collection of tweets with sarcasm tags. Finally, the classification results have evaluated using different classification measures such as accuracy, precision, recall, and F-measure.

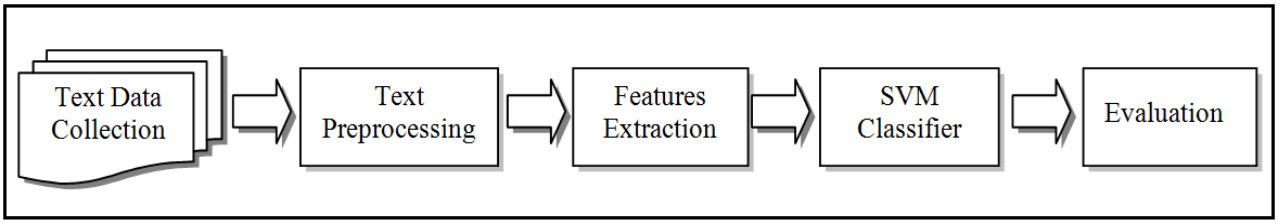

Fig. 1. The various stages of our proposed approach.

A. Text data collection

This is the first stage of the approach where the text corpus is collected. One of the common ways of getting a text corpus is to use the online social networking service; Twitter to download a large amount of tweets. We decided to work with Twitter due to the popularity of social networking websites and the ability to extract a lot of useful information from what people post on such websites.

To build our corpus of sarcastic and non sarcastic tweets, we relied on the annotations that tweeters assign to their own tweets using hashtags. Our assumption is that the best judge of whether a tweet is intended to be sarcastic or not is the author of the tweet. Furthermore, using the tweets labeled by their authors using hashtag produces a better quality gold standard and it allows for the creation of large-scale datasets. Using hashtags to select sarcastic and non-sarcastic tweets is known as automatic annotation. Another way of doing it is through manual annotation.

To obtain a set of sarcastic tweets, we used a search query to collect all the tweets that posted during the years 2010 to 2020 using a set of predefined hashtags that express sarcasm, including #سخرية, #سخريه, #مسخرة, #مسخره, #تهكم, #استهزاء, #ساركازم (All of these words are synonyms meaning #sarcasm) and setting the language parameter to Arabic to make sure

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 8, August 2021
DOI 10.17148/IJARCCE.2021.10801

that the search only pulls in Arabic tweets. To ensure quality, these tweets are manually investigated and annotated by us.

As for non sarcastic tweets, we collected tweets for some particular domains such as politics, and sports in the same manner and made sure they have some emotional content. We have chosen these two domains because we found that the most of the tweets contents that collected in the sarcastic set is belonging to politics and sports domains. Hence, the tweets of the two sets will have similar content, similar style, but different intentions. The non satrcastic tweets were retreived by searching with hashtags such as #سياسة, #سياسه, #رياضة, #رياضه (translation of #politics and #sports respectively) and setting the language parameter to Arabic to make sure that the search only pulls in Arabic tweets. It is possible, and also very likely, that the non sarcastic set also contain some sarcastic tweets even though no #sarcasm hashtag has been used. We assume though that the number of sarcastic tweets in the non sarcastic set is relatively small and thus negligible.

The collection process resulted in a set of 20000 tweets (10000 are sarcastic tweets and 10000 are non sarcastic) as depicted in Table I. The collected corpus consists of tweets written in Modern Standard Arabic (MSA) and Dialectical Arabic (DA) or a a mix of MSA and DA. The corpus will be made freely available for research purposes.

TABLE I. OVERVIEW OF THE CORPUS WITH SARCASTIC AND NON SARCASTIC TWEETS

Class	Hashtags	#Instances
Sarcasm	#سخرية, #سخريه, #مسخرة, #مسخره, #تهكم, #استهزاء, #ساركازم	10000
Not Sarcasm	#سياسة, #سياسه, #رياضة, #رياضه	10000

B. Text preprocessing

Some preprocessing in the Arabic text corpus have performed. It includes cleaning text, tokenizing string to words, applying stopwords removal, applying the suitable term stemming and pruning methods as a feature reduction. We used the open source machine learning tool RapidMiner for text preprocessing.

1) Text cleaning: This step includes eliminating the irrelevant contents in the tweets such as user names, hashtags, URLs, emails, reference tweets, since they are weakly informative in sarcasm detection.

2) Tokenization: It is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such text mining [21].

3) Stopwords removal: Stopwords are terms that are too frequent in the text. These terms are insignificant and do not have meaning or do not hold information. For Arabic, stopwords list includes articles, conjunctions, prepositions, pronouns, days of week, and months of year [22].

4) Stemming: For Arabic Language, there are two different morphological analysis techniques; root-based stemming and light stemming. Root-based stemming removes all affixes and returns each inputted Arabic word to its root pattern. Whilst, light stemming eliminates only the common affixes (prefixes and suffixes) without altering the origin (root) of a word [7]. In this study, we apply the light stemming algorithm. The main reason for this choice is that many words which share the same root have completely different meanings. Thus, we maintain the correct meaning of the word. Furthermore, the light stemming is more proper than stemming from linguistics and semantic view point, and it has the least preprocessing time, it also has superior average classification accuracy.

5) Pruning: It is the process of eliminating the words that its count is less or greater than a specific threshold to reduce the dimensionality of text data. This step is necessary to save storage and time when we classify a corpus [23]. For avoiding an unnecessary large dimension space only words that occur more than 10 times in the corpus are considered as features.

C. Features extraction

The task of converting a given text into a feature vector is an important task in text processing in terms of extracting the most important features.

1) Vector Space Model (VSM): Before using SVM classifier on the data, we need to represent the text in a format suitable for the classifier to deal with it. In NLP, the popular model is the Vector Space Model (VSM) or feature vector. It represents documents as vectors in m-dimensional space. The text, either a document or a sentence, will be converted into the form of the features model before the training process of the classifier starts. This model should preserve essential information about the text. Each row of the model represents one of the data set records (either document or sentence). Each column displays the features that are chosen to build the vector model. The intersection of each row with each column contains a value that represents the relation of that feature in that data record [14].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

2) Term weighting: The aim of term weighting is to enhance text document representation as feature vector or VSM. Popular term weighting schemes are Binary Term Occurrences (BTO), Term Occurrences (TO), Term Frequency (TF), and Term Frequency-Inverse Document Frequency (TF-IDF). BTO indicates absence or presence of a word with Booleans 0 or 1 respectively. TO is the number of occurrences of term t in the document d.Term frequency TF(t, d) is the number that the term t occurred in the document d. Document Frequency DF(t) is number of documents in which the term t occur at least once. The inverse document frequency can be calculated from document frequency using the formula log(num of Docs/num of Docs with word i). The inverse document frequency of a term is low if it occurs in many documents and high if the term occurs in only few documents. Term discrimination consideration suggests that the best terms for document content identification are those able to distinguish certain individual documents from the collection. This implies that the best terms should have high term frequencies but low overall collection frequencies (num of Docs with word i). A reasonable measure of term importance may then be obtained by using the product of the term frequency and the inverse document frequency (TF * IDF). The TF-IDF is a statistical measure used to evaluate how important a word is to a document in a collection or corpus [24, 25, 26]. In this work, we applied several term weighting schemes, including TF, TF-IDF, BTO, and TO.

3) N-gram features: A good set of features is the N-gram features. The N-gram model is sometimes called the Bag of Words (BOW) model. It refers to a sequence of words within a tweet, where N indicates the size (number of words) of a sequence. The common used sizes of N-gram are uni-gram (N = 1), bi-gram (N = 2) and tri-gram (N = 3). [18]. In the case of the uni-gram model, the feature of BOW will contain only one word from the distinct words of the corpus. In bi-gram, tri-gram, and etc. models,the feature will contain a combination of two, three, or n words depending on the model type [14]. Uni-gram provide a good coverage of the data set, while bi-gram and tri-gram provide the ability to capture some relationships and dependencies between the words and to capture the effect of individual phrases on sarcasm.[14, 27].To represent the candidate features, we use several term weighting schemes, including TF, TF-IDF, BTO, and TO on the three N-gram models; uni-gram, bi-gram, and tri-gram. For avoiding an unnecessary large dimension space only words that occur more than 10 times in the corpus are considered as features. We used the open source machine learning tool RapidMiner for N-gram features extraction and representation.

D. Support Vector Machines (SVM) classifier

Support Vector Machine (SVM): SVM [28] is supervised machine learning algorithm that aims to find the "maximal margin" hyperplane that separates the classes. This hyperplane is defined by the support vectors (examples near the class boundaries) so that the distance between these support vectors of different classes is maximized as shown in Fig. 2.

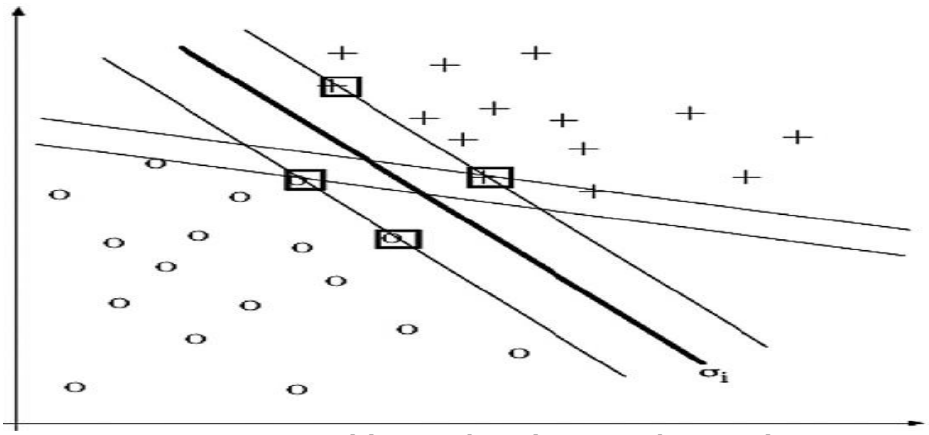

Fig. 2. Support vectors delimiting the widest margin between classes.

For the tweet classification task, we choose the SVM algorithm under the RapidMiner tool to judge whether the given tweet is sarcastic or not. This algorithm were chosen due to its simplicity and effectiveness in many text classification tasks. We fed the SVM algorithm with the feature vectors generated from the tweets data to train the classifier, so that the classifier would build the classification model that new data could be classified according to it. We use the linear kernel to perform the classification task because it does not consume as much time and resources on a large amount of data as polynomial kernel.

E. Evaluation

Certain metrics are needed to evaluate the classifiers performance. A common way to evaluate the performance of the classifiers is by using a confusion matrix. Confusion matrix (also callled a performance vector) is a useful tool for analyzing how well your classifier can recognize data. It contains information about realistic and predicted classifications [29]. From the confusion matrix, we can calculate the performance using four measures, including accuracy, precision, recall, and F-measure which are generally accepted ways of measuring system's success in this field. These measures are defined as follows [29]:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

- Accuracy: it represents the overall correctness of classification. In other words, itmeasures the fraction of all correctly classified instances over the total number of instances.
- Precision: it represents the fraction of retrieved sarcastic tweets that are relevant. In other words, it measures the number of tweets that have successfully been classified as sarcastic over the total number of tweets classified as sarcastic.
- Recall: it represents the fraction of relevant sarcastic tweets that are retrieved. In other words, it measures the number of tweets that have successfully been classified as sarcastic over the total number of sarcastic tweets.
- F-measure: it is a standard statistical measure that is used to measure the performance of a classifier system. The F-measure is an average parameter based on precision and recall.

IV. EXPERIMENTS AND RESULTS

This section illustrates the experiments that have been performed to investigate and test the features and performance of the SVM classifier on sarcasm detection. It presents the experimental results and their evaluation. It also discusses the obtained results to justify our proposed approach feasibility.

A. Experimental setup

In this subsection, we describe the experimental process we have used to evaluate our approach for the task of identifying sarcastic tweets.

For the tweet classification task experiments, we have used the corpus that we built in order to apply the SVM classifier for the problem of sarcasm detection in Arabic tweets. Our corpus consists of a total of 20000 tweets (10000 are sarcastic tweets and 10000 are non sarcastic). We split the corpus into 2 sets (70% of the corpus for training and the remaining 30% for testing) as depicted in Table II. The training set is used to build the model and the test set is used to validate it. We used stratified sampling to build the sets, it ensures that the class distribution in the sets is the same as in the whole corpus.

TABLE II. TWEETS DISTRIBUTION IN OUR CORPUS

	Sarcasm	Not Sarcasm	Total
Training	7000	7000	14000
Testing	3000	3000	6000
Total	10000	10000	20000

Using the tweets, we construct N-gram features. This feature extraction process produces feature vectors used to train SVM algorithm which produces a classifier model classifying tweets as sarcastic or non satrcastic.

To carry out the experimentation, we have used SVM classifier under the RapidMiner tool to judge whether the given tweet is sarcastic or not. This classifier were chosen because they have shown best results in many text classification tasks.

We have carried out a two different groups of experiments in order to evaluate the effectiveness of our approach, in automatically distinguishing between sarcastic and non sarcastic tweets over the collected corpus. These experiments are grouped according to N-gram features, including: uni-gram features, and a various combinations of uni-gram, bi-gram and tri-gram features; in each group, the SVM classifier has been applied in order to determine the best set of features and the best term weighting schemes, including TF, TF-IDF, BTO, and TO used to represent these features. The details of these experiments and their results are explained in the next subsection.

One of the primary goals of this work is to evaluate the different feature sets as well as different term weighting schemes, including TF, TF-IDF, BTO, and TO used to represent these features for recognizing the sarcasm in the Arabic text of Twitter data. In order to do that, we need to establish a proper baseline experiment before starting to do comparison experiments. This provides a useful method to compare the performance of different term weighting schemes with the corresponding feature sets. The question here is what the best baseline is. It is hard to judge or make the optimal baseline experiment because the baseline will vary depending on the nature of the task. In our case, we choose the unigram model, as the baseline that provides the point of reference for judging other feature set experiments for each classifier. This baseline might be fair because it preserves the basic knowledge about the text classification problem which is the general topic of sarcasm detection.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

To evaluate the classifiers, we rely on calculating accuracy, precision, recall, and F-measure which are generally accepted ways of measuring system's success in this field. In order to compute these metrics, the confusion matrix should be generated after the classification process.

B. Experimental results and discussion

This subection presents and discusses the results of the numerous experiments that have been conducted. The main idea behind these experiments was to establish the best feature sets and term weighting schemes used to represent these features that work well for sarcasm detection in Arabic text.

1) Experiments with the uni-gram features (Baseline experiments)

In these experiments, we try to establish baseline results so that we can compare our next experiments according to that. The baseline that we choose is the one that provides the basic knowledge about the text and might preserve the primary semantic feature of the language. Therefore, we use the uni-gram feature model as a baseline model. We performed these experiments using SVM classifier with different term weighting schemes, including TF, TF-IDF, BTO, and TO; four experiments have been done. This feature set has 3559 distinct feature.

Table III displays the four classification performance measures of the baseline experiments for the four term weighting schemes using SVM classifier. The values in the table refer to the accuracy, precision, recall, and F-Measure that is calculated after performing split validation. The bolder values indicate the best results for the term weighting schemes for the baseline experiments.

TABLE III. CLASSIFICATION PERFORMANCE MEAURES FOR THE FOUR TERM WEIGHTING SCHEMES OF THE BASELINE EXPERIMENTS

	TF			
	Accuracy	**Precision**	**Recall**	**F-measure**
Unigram Baseline	**86.25%**	**86.46%**	**86.25%**	**86.35%**
	TF-IDF			
	Accuracy	**Precision**	**Recall**	**F-measure**
Unigram Basline	86.15%	86.39%	86.15%	86.27%
	BTO			
	Accuracy	**Precision**	**Recall**	**F-measure**
Unigram Baseline	85.75%	86.31%	85.75%	86.03%
	TO			
	Accuracy	**Precision**	**Recall**	**F-measure**
Unigram Baseline	85.02%	85.76%	85.02%	85.39%

Fig. 3 gives a graphical summary of the four classification performance measures of the baseline experiments for the four term weighting schemes using SVM classifier.

As can be seen from Fig. 3, The highest overall accuracies obtained with SVM classifier reach 86.25% and 86.15% using the TF and TF-IDF term weighting schemes respectively for an f-measure equal to 86.35% and 86.27%. The BTO, and TO term weighting schemes have a comparable performance on the task with a slight difference, which is albeit lower compared to the performance of the TF and TF-IDF term weighting schemes. This indicates that the machine learning approach using SVM classifier is appropriate for the identification of sarcasm. In addition, This suggests that the baseline lexical features are informative in the sarcasm identificatrion task.

IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

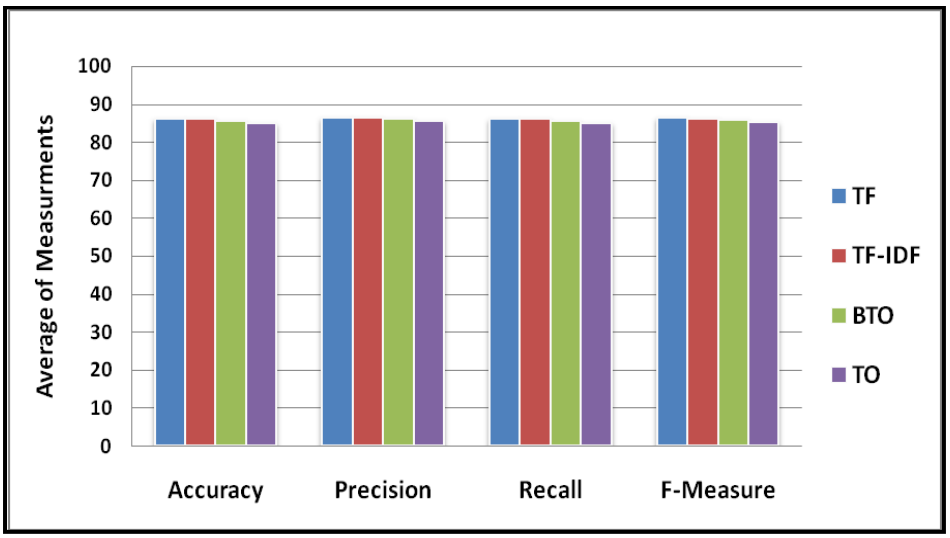

Fig. 3. Classification performance meaures for the four term weighting schemes of the baseline experiments using SVM classifier.

In the following experiments, we will use these results as a baseline to help us in judging the different feature sets. In addition, it gives evidence whether the classifier could learn new knowledge from the extra features.

2) Experiments with combinations of different N-gram features

In these experiments, we try to compare between the various feature combinations using the different N-gram features, including uni-gram, bi-gram and tri-gram features to investigate their effect on the SVM classifier with different term weighting schemes for Arabic sarcasm detection. The initial experiments that have been done in the earlier preparation stages of this work indicate that using different N-gram such as bi or tri-gram alone do not increase the accuracy of the classification. Therefore, we only consider adding the different N-grams to the uni-gram model which is the baseline model.

In order to perform these experiments, we form three feture sets from various N-gram features combinations. The first feature set is using the uni-gram model with the bi-gram model, we add 891 bi-gram feature to our uni-gram baseline. This resulted in a set of 4450 feature. The second feature set is using the uni-gram model with the tri-gram model, we add 34 tri-gram feature to our uni-gram baseline. This resulted in a set of 3593 feature. The third and last feature set feature set is using the uni-gram model with the bi-gram and tri-gram models, we add 891 bi-gram feture and 34 tri-gram feature to our uni-gram baseline. This resulted in a set of 4484 feature.

The primary goal of these experiments is to find the N-gram model combination that works best with SVM classifier using different term weighting schemes for the Arabic text. The second goal is to figure out if the N-gram model could capture some of the relationships between the words and capture the effect of individual phrases on sarcasm, especially in the case of bi and tri-gram model. We performed these experiments using SVM classifier with different term weighting schemes, including TF, TF-IDF, BTO, and TO; 12 experiment have been done.

Table IV displays the four classification performance measures of the four term weighting schemes using SVM classifier for the baseline experiments and for the experiments of the three feture sets that formed from various N-gram feature combinations. The values in the table refer to the accuracy, precision, recall, and F-Measure that is calculated after performing split validation. The bolder values indicate the best results achieved among all feature sets and term weighting schemes using SVM classifier. The underlined values indicate the best results that are achieved using a particular feature model for each term weighting scheme using SVM classifier.

TABLE IV. CLASSIFICATION PERFORMANCE MEAURES FOR THE FOUR TERM WEIGHTING SCHEMES OF THE BASELINE AND VARIOUS N-GRAM FEATURE COMBINATIONS EXPERIMENTS

	TF			
	Accuracy	**Precision**	**Recall**	**F-measure**
Unigram Baseline	86.25%	86.46%	86.25%	86.35%
Unigram + Bigram	**86.60%**	**86.72%**	**86.6%**	**86.66%**
Unigram + Trigram	86.33%	86.52%	86.34%	86.43%

IJARCCE
ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 8, August 2021
DOI 10.17148/IJARCCE.2021.10801

Unigram + Bigram +Trigram	**86.60%**	**86.72%**	**86.6%**	**86.66%**
	TF-IDF			
	Accuracy	**Precision**	**Recall**	**F-measure**
Unigram Baseline	86.15%	86.39%	86.15%	86.27%
Unigram + Bigram	86.20%	86.4%	86.2%	86.30%
Unigram + Trigram	85.97%	86.22%	85.97%	86.09%
Unigram + Bigram +Trigram	86.22%	86.42%	86.22%	86.32%
	BTO			
	Accuracy	**Precision**	**Recall**	**F-measure**
Unigram Baseline	85.75%	86.31%	85.75%	86.03%
Unigram + Bigram	85.93%	86.39%	85.94%	86.16%
Unigram + Trigram	85.63%	86.22%	85.64%	85.93%
Unigram + Bigram +Trigram	85.95%	86.40%	85.95%	86.17%
	TO			
	Accuracy	**Precision**	**Recall**	**F-measure**
Unigram Baseline	85.02%	85.76%	85.02%	85.39%
Unigram + Bigram	85.60%	86.21%	85.6%	85.90%
Unigram + Trigram	84.90%	85.65%	84.9%	85.27%
Unigram + Bigram +Trigram	85.55%	86.17%	85.55%	85.86%

Fig. 4 gives a graphical summary of the four classification performance measures of the experiments with the unigram + bigram feature set for the four term weighting schemes using SVM classifier.

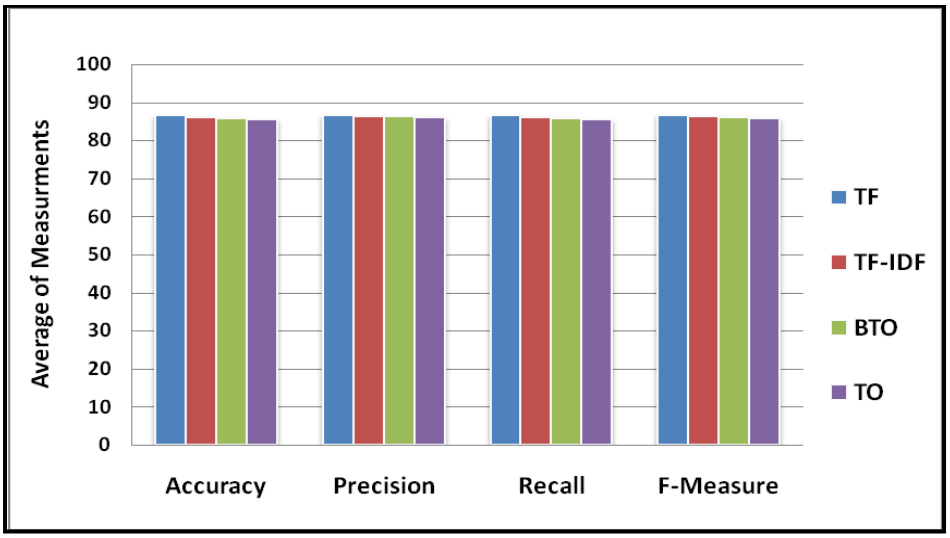

Fig. 4. Classification performance meaures for the four term weighting schemes of the experiments with the unigram + bigram feature set using SVM classifier.

As can be seen from Fig. 4, The highest overall accuracies obtained with SVM classifier reach 86.60% and 86.20% using the TF and TF-IDF term weighting schemes respectively for an f-measure equal to 86.66% and 86.30%. The BTO, and TO term weighting schemes have a comparable performance on the task with a slight difference, which is albeit lower compared to the performance of the TF and TF-IDF term weighting schemes.

IJARCCE
ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 8, August 2021
DOI 10.17148/IJARCCE.2021.10801

Fig. 5 gives a graphical summary of the four classification performance measures of the experiments with the unigram + trigram feature set for the four term weighting schemes using SVM classifier.

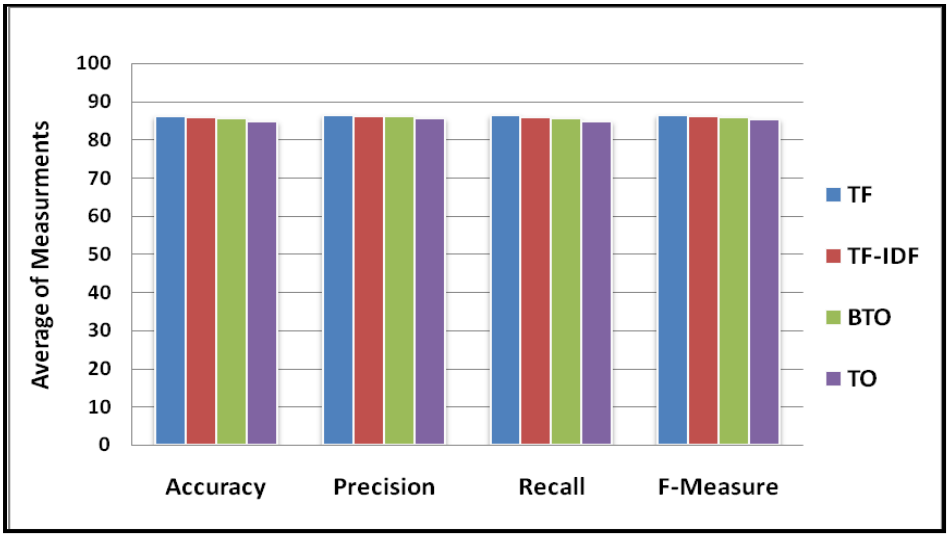

Fig. 5. Classification performance meaures for the four term weighting schemes of the experiments with the unigram + trigram feature set using SVM classifier.

As can be seen from Fig. 5, The highest overall accuracies obtained with SVM classifier reach 86.33% and 85.97% using the TF and TF-IDF term weighting schemes respectively for an f-measure equal to 86.43% and 86.09%. The BTO, and TO term weighting schemes have a comparable performance on the task with a slight difference, which is albeit lower compared to the performance of the TF and TF-IDF term weighting schemes.

Fig. 6 gives a graphical summary of the four classification performance measures of the experiments with the unigram + bigram + trigram feature set for the four term weighting schemes using SVM classifier.

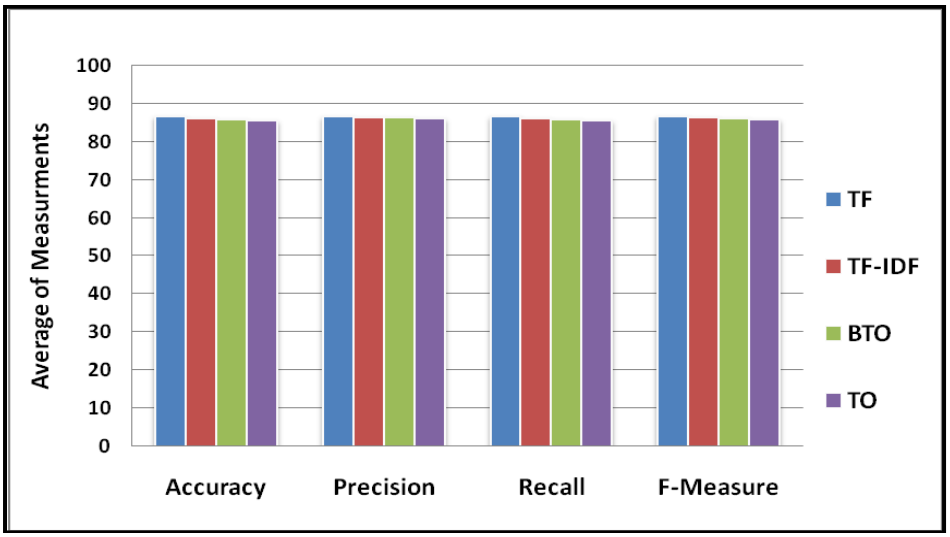

Fig. 6. Classification performance meaures for the four term weighting schemes of the experiments with the unigram + bigram + trigram feature set using SVM classifier.

As can be seen from Fig. 6, The highest overall accuracies obtained with SVM classifier reach 86.60% and 86.22% using the TF and TF-IDF term weighting schemes respectively for an f-measure equal to 86.66% and 86.32%. The BTO, and TO term weighting schemes have a comparable performance on the task with a slight difference, which is albeit lower compared to the performance of the TF and TF-IDF term weighting schemes.

It is clear from Table IV and Figures 4,5, and 6 that the TF term weighting scheme achieves the best results for all feature sets among other term weighting schemes using SVM classifier.

The best results among all feature sets and term weighting schemes achieved using the TF term weighting scheme with the unigram + bigram feature set and with the unigram + bigram + trigram feature set with the overall accuracy equal to 86.60% using SVM classifier which these results are quite high especially regarding Arabic text. The TF-IDF term

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 8, August 2021
DOI 10.17148/IJARCCE.2021.10801

weighting scheme also showed a significant increase in the performance and achieved accuracy and f-measure equal to 86.22% with the unigram + bigram + trigram feature set. It can be noticed that adding biagram, bigram and trigram features to the unigram baseline model improved the classification. It helps to improve the results of all term weighting schemes using SVM classifier. This improvement might be coming from using bi- and tri-gram that capture some relationship between words. This also suggests that these combined features are coherent. The highest obtained results indicate that the machine learning approach using SVM classifier is appropriate for the identification of sarcasm in Arabic language. In addition, This suggests that various feature combinations using the different N-gram features are powerful and informative in the sarcasm identificatrion task.

V. CONCLUSION

In this paper, a proposed approach for detecting sarcasm in the Arabic tweets has been introduced. Various stages are involved in this approach, including text data collection, text preprocessing, features extraction, SVM classification, and evaluation. The corpus were used for the experiments carried out in this research is collected from the online social networking service; Twitter. This corpus comprises of 20000 tweets (10000 are sarcastic tweets and 10000 are non sarcastic). The collected corpus consists of tweets written in MSA and DA or a a mix of MSA and DA.

We have conducted several experiments by applying the SVM classifier to classify sarcastic tweets. These experiments are grouped according to N-gram features, including: uni-gram features, and a various combinations of uni-gram, bi-gram and tri-gram features; in each group, the SVM classifier has been applied in order to investigate the best set of features and the best term weighting schemes, including TF, TF-IDF, BTO, and TO used to represent these features.The accuracy, precision, recall, and f-measure are used for the evaluation of the classifiers performance.The experiments gave promising results. The best results by SVM classifier for all feature sets and several weighting schemes achieved overall accuracy equal to 86.60%, which these results are quite high especially regarding Arabic text.

The field of Arabic sarcasm detection is still in an early stage. Therefore, there are many different areas and directions of improvement and future investigation for Arabic sarcasm detection as follows:

- We evaluated our approach on a balanced dataset, where the number of sarcastic and nonsarcastic tweets are equal. However, this situation rarely occurs in a real situation, since the number of non-sarcastic tweets may be much higher than the number of sarcastic tweets. We also need to evaluate our method on an unbalanced dataset and a real dataset.
- Evaluating the performance of the proposed approach on a large corpus and other domains.

REFERENCES

[1] Kaplan, A. and Haenlein, M. (2010) 'Users of the world, unite! The challenges and opportunities of social media', Business Horizons, vol. 53, no. 1, pp. 59-68.

[2] Aldayel, H. and Azmi, A. (2015) 'Arabic tweets sentiment analysis – a hybrid scheme', Journal of Information Science, vol. 42, no. 6, pp. 782 - 797.

[3] P´erez, A. (2012) 'Linguistic-based Patterns for Figurative Language Processing: The Case of Humor Recognition and Irony Detection', PhD. Dissertation, Information Systems and Computing Department, Polytechnic University of Valencia.

[4] Facebook official website, Available: https://about.facebook.com/ [Online; accessed 13 January 2021].

[5] Twitter official website, Available: http://about.twitter.com/company [Online; accessed 8 January 2021].

[6] Weitzel, L., Prati, R. and Aguiar, F. (2016) 'The Comprehension of Figurative Language: What Is the Influence of Irony and Sarcasm on NLP Techniques?', Sentiment Analysis and Ontology Engineering, book chapter, pp. 49 -74, Springer.

[7] Abdulla, N., Ahmed, N., Shehab, M., Al-Ayyoub, M., Al-Kabi, M. and Al-rifai, S. (2014) 'Towards Improving the Lexicon-Based Approach for Arabic Sentiment Analysis', International Journal of Information Technology and Web Engineering, vol. 9, no. 3, pp. 55-71.

[8] El-Makky, N., Nagi, K., El-Ebshihy, A., Apady, E., Hafez, O., Mostafa, S., and Ibrahim, S. (2015). 'Sentiment Analysis of Colloquial Arabic Tweets', The 3rd ASE International Conference on Social Informatics (SocialInformatics 2014) – Conference Proceedings, MA, USA.

[9] Oxford dictionary, 'Definition of Sarcasm', Available: http://en.oxforddictionaries.com/definition/sarcasm [Online; accessed 9 January 2021].

[10] Merriam-Webster dictionary, 'Definition of Sarcasm', Available: http://www.merriam-webster.com/dictionary/sarcasm [Online; accessed 14 January 2021].

[11] Kunneman, F., Liebrecht, C., Van Mulken, M. and Van Den Bosch, A. (2014) 'Signaling Sarcasm: From Hyperbole to Hashtag', Information Processing and Management Journal, vol. 51, no. 4, pp. 500-509.

[12] Gibbs, R. and Colston, H. (2007) 'Irony in Language and Thought', 1st edition, Routledge (Taylor and Francis), New York.

[13] Wikipedia, 'The Arabic language', Available: https://en.wikipedia.org/wiki/Arabic [Online; accessed 11 January 2021].

[14] Alotaibi, S. (2015) 'Sentiment Analysis in the Arabic Language Using Machine Learning', PhD. Dissertation, Department of Computer Science, Colorado State University.

[15] Carvalho, P., Sarmento, L., Silva, M. and Oliveira, E. (2009) 'Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-)', The 1st international Conference on Information and Knowledge Management workshop on Topic-sentiment analysis for mass opinion – Conference Proceedings, Hong Kong, China, pp. 53-56.

[16] Liebrecht, C., Kunneman, F. and Van Den Bosch, A. (2013) 'The Perfect Solution for Detecting Sarcasm in Tweets #Not', The 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis – Conference Proceedings, Atlanta, Georgia, pp. 29-37.

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 8, August 2021

DOI 10.17148/IJARCCE.2021.10801

[17] Riloff, E., Qadir, A., Surve, P. and De Silva, L. (2013) 'Sarcasm as Contrast between a Positive Sentiment and Negative Situation', The 2013 Conference on Empirical Methods in Natural Language Processing – Conference Proceedings, Seattle, Washington, USA, pp. 18-21.

[18] Tungthamthiti, P., Shirai, K. and Mohd, M. (2014) 'Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches', The 28th Pacific Asia Conference on Language, Information and Computation – Conference Proceedings, Phuket, Thailand, pp. 404-413.

[19] Bharti, S., Babu, K. and Jena, S. (2015) 'Parsing-based Sarcasm Sentiment Recognition in Twitter Data', The 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining – Conference - International, Paris, France, pp. 1373-1380.

[20] Bouazizi, M. and Ohtsuki, T. (2015) 'Sarcasm Detection in Twitter "All your products are incredibly amazing!!!" – Are They Really ? ', The 2015 IEEE Global Communications Conference (GLOBECOM 2015) – Conference - International, San Diego, CA, USA, pp. 1-6, December 06-10.

[21] Wikipedia, 'Tokenization', Available: https://en.wikipedia.org/wiki/Lexical_analysis#Tokenization [Online; accessed 19 January 2021].

[22] Feldman, R. and Sanger, J. (2007) 'The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data', Cambridge University Press.

[23] Saad, M. (2010) 'The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification', M.Sc. Dissertation, Department of Computer Engineering, The Islamic University-Gaza.

[24] Jing, L., Huang, H. and Shi, H. (2002) 'Improved feature selection approach TFIDF in text mining', The International Conference on Machine Learning and Cybernetics – Conference Proceedings, Beijing, China, vol. 2, pp. 944-946, November 4-5.

[25] Said, D., Wanas, N., Darwish, N. and Hegazy, N. (2009) 'A Study of Arabic Text preprocessing methods for Text Categorization', The Second International Conference on Arabic Language Resources and Tools – Conference Proceedings, Cairo, Egypt.

[26] Salton, G. and Buckley, C. (1998) 'Term weighting approaches in automatic text retrieval', Information Processing and Management, vol. 24, no. 5, pp. 513-523.

[27] Saleh, M., Martín-Valdivia, M., Ureña-López, L. and Perea-Ortega J. (2011) 'OCA: Opinion Corpus for Arabic', Journal American Society for Information Science and Technology, vol. 62, no. 10, pp. 2045–2054.

[28] Hearst, M. (1998) 'Trends & controversies: Support vector machines', IEEE Intelligent Systems, vol. 13, no. 4, pp. 18-28.

[29] Han, J. and Kamber, M. (2006) 'Data Mining: Concepts and Techniques', 2nd edition. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor, San Francisco, USA.

