The Effect of Feature Selection on Classifying Arabic Comment Polarity

Hana A. Al-Hatibi[1], Adel Sallam M. Haider[1], Abdulla A. Omeer[2]
[1]Department of Information Technology, Faculty of Engineering ,University of Aden, Aden, Republic of Yemen
[2]Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, MH, India
hana_hatibi@yahoo.com[1], adel_ye@yahoo.com[2], binomeer1979@yahoo.com[3]

Abstract— *This paper investigates implement Machine Learning Approaches for Sentiment analysis (SA) to grouping text documents into positive or negative polarity with respect to their contents. In this paper, we highlight two problems for Arabic Sentiment analysis (ASA) technique**:** First, the Preprocessing phase of representing each document as a feature vector, that is, to separate the text into individual words. Second compare the performance between tree famous classifiers Naïve Bayes (NB), K-Nearest Neighbor (KNN, with k=15 and Jaccard measure) and Support Vector Machine (SVM with Nova kernel and C=10) learning algorithms used to solve the automatic text classification task and determine the class of the document is positive or negative polarity, Our results on OCA corpus which available resources composed of Arabic language ,we used Pruning perceptual method with threshold(minimum threshold of 3% and maximum of 30%)and Information Gain IG as feature selection, before building model which ignores words that appear in less/more than this percentage of all documents, preprocessed the corpus (tokenization, stemming, and stop word removal) and calculating the weights of each feature (TFIDF), Experimental results showed that applying different parameters of learning algorithms impact on the performance ,we perform some comparative experiments between machine learning algorithms, From the experimental results, it can be concluded that the combination feature selection IG and threshold achieves the best performance for tree classifier*.

Keywords— Sentiment Analysis, K-NN, NB, SVM, TFIDF and IG

I. INTRODUCTION

Nowadays web applications and social media have rapid growth, there became reviews, comments, ratings and feedback about several varied issues generated by users, All of which need to be analyzed to get a good estimation of what the user thinks and feels [1]. Before the availability of automatic sentiment analysis tools, the process of obtaining customers' reviews was an extremely cumbersome and time-consuming process. This probably explains the great interest of this field of research.

Sentiment analysis (SA), opinion mining or sentiment orientation is various names in the literature to assign an opinionated review to one or more categories, with respect to their contents**. i.e**. to grouping review or comment into positive or negative polarity [2], Sentiment analysis has three different levels such as document level, sentence level, and aspect level [3]. Sentiment-based categorization in the movie review is document-level sentiment analysis (SA). It treats the review as a set of number of independent words by ignoring the sequence of words on a text. Every single unique word and phrase can be used as the document features. In this paper, the dataset is chosen for system implementation are the movie-review corpus and compare tree Machine learning algorithms on document level to determine attitude polarity of phrases embedded in comments.

In this paper, we conducted an experiment to prove that higher performance, using different algorithm parameters can impact the Sentiment Analysis performance. As far as quality and performance are concerned, we used RapidMiner which widely used tool for researches that develop machine learning classifiers.

This paper is organized as follows: Section I has provided an introduction to this paper. Section II Related Work, Section III Methodology. Section IV Evaluation parameter, Section V Experiment Result and finally section VI summarizing the conclusions of this paper.

II. RELATED WORK

In the field of Arabic sentiment analysis research many works were done for comparing classifiers on Arabic texts.

Rehab M. Duwairi [2] utilize machine learning techniques to determine the polarity of tweets written in Arabic with the presence of dialects, and collected and annotated a dataset of Arabic tweets which consists of 22550 tweets was gathered using Twitter API and annotated using the Crowdsourcing Tool two classifiers, namely: the Naïve Bayes (NB) and the Support Vector Machine (SVM) classifiers were used to determine the polarity of the tweets. These classifiers built their classification models by using two versions of the same dataset. The first version consists of tweets that contain dialectical words and the second version consists of tweets after translating the dialectical words into their corresponding MSA words by utilizing a dialect lexicon.

Nawaf A. Abdulla et al [4] compares the accuracy of a corpus-based and a lexicon-based techniques for SA, the authors classify the whole document into one of the three polarity classes(positive, negative and neutral) via both supervised and unsupervised approaches, they collected an Arabic dataset composed of 2000 Tweets and build an Arabic lexicon from a seed list of 300 words, The preprocessing steps applied on the dataset included spelling correction, elongation , stop-word removal, and letter normalization. The authors experimented with a set of classifiers including Support Vector Machines (SVM), Naive Bayes (NB), and K-Nearest Neighbor (KNN) with K=9, using RapidMiner. The best results were reported to be those of SVM and NB (using 5-fold cross-validation and light stemming) with accuracies of 87.2% and 81.3% respectively.

Sasi Atia et. al. [5] used Opinion Corpus for Arabic (OCA) with NB and SVM classifiers, achieves higher Accuracy than the baseline (Rushdi, Martin & Alfonso 2011) [6], and achieved the highest accuracy when used Binary Term Occurrence (BTO) for NB and the SVM machine learning technique achieved the highest accuracy level when the Term Frequency- Inverse Document Frequency TFIDF.

Dina Said et al. [7] studies difference tools for Arabic text preprocessing, attribute selection and reduction and classification, The results illustrated that using light stemmer combined with a good performing feature selection method enhances the performance of Arabic Text Categorization, Two datasets are used in this study which is Aljazeera News Arabic Dataset **(**1500 Arabic news documents**)** and Al-jazirah Magazine Arabic Dataset (4470 articles).

Motaz et al**.** [8] study the impact of text pre-processing and different term weighting schemes on Arabic text classification collected manually from Aljazeera news web site, The dataset contains 119 text documents belonging to one of the three categories (sport, health, computer & communications) , use *C4.5* decision tree with 10 folds cross-validation. The result shows that Stemming enhances term weighting and this affect classification accuracy.

Duwairi et al**.** [9] investigated the effects of stemming, feature correlation, and n-gram models for Arabic text on sentiment analysis, the behavior of three classifiers, namely, SVM, Naïve Bayes, and K-nearest neighbor classifiers and the effects of the characteristics of the dataset on sentiment analysis, Two datasets were used. One dataset is called the Politics dataset and it consists of 300 reviews: 164 positive reviews and 136 negative reviews. These reviews were collected by the authors of this paper from the Aljazeera website. The other dataset, by comparison, is called the Movie dataset and is publically available. It consists of 500 reviews: 250positive reviews and 250 negative reviews. The results show that selection of preprocessing strategies on the reviews increases the performance of the classifiers.

Duwairi et al. [10] compared three dimensionality reduction techniques; stemming, light stemming, and word cluster. The authors used *KNN* to perform the comparison. Performance metrics are: time, accuracy, and the size of the vector. She showed that light stemming is the best in terms of classification accuracy. Duwairi collected 1,500 documents belonging to one of three categories (sport, economic, education). Each category has 5,000 documents. She split the corpus; 9,000 documents for training and 6,000 documents for testing.

Monica et al. [11] study feature selection methods for text classification on Two benchmark collections were chosen as the testbeds: Reuters-21578 and small portion of Reuters Corpus Version 1 (RCV1), they found that feature selection methods based on $\chi 2$ statistics consistently outperformed those based on other criteria (including information gain) for all of the four classifiers ((NB, KNN, Rocchio-style classifier and SVM) and both data collections.

Asriyanti [12] investigate Sentiment analysis in a movie review and proposed feature selection; IGDFFS selects sub-features that satisfy these criteria: (1) high relevance to the output class and (2) high occurrence in the dataset. As a result, it constructs sub-features that reach better performance in the classification.

III. METHODOLOGY

In this section, we propose a general architecture, Fig.1 of our approach which is performed through the following five main steps:

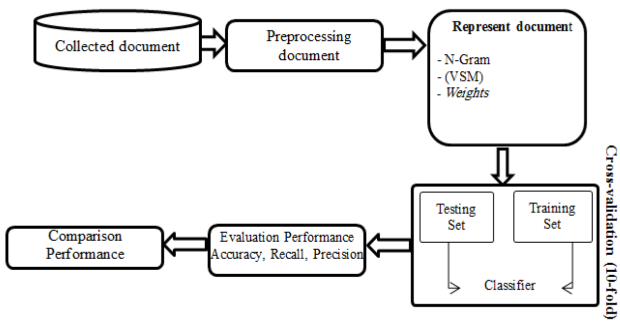

Fig.1: The framework of our experiments

1. Dataset

Table (1) shows collected dataset. The dataset is chosen for system implementation was the Opinion Corpus for Arabic (OCA) movie-review corpus sites by Saleh et al. [6], the corpus consists of 500 Arabic movie reviews, 250 positive and 250 negative, collected from 15 different Arabic web sites. Each of the written reviews is sets of text detailing the movie and author's commentary, this corpus is freely available at [13], they used the RapidMiner machine learning tool to conduct to build a classifier. The reported accuracy results were 0.8900 for NB and 0.9060 for SVM.

TABLE 1: DATASET

Dataset	Positive	Negative
Total Document	250	250
Total Token	94,556	121,392

2. Text Pre-processing

Sentiment analysis goes through a number of stages. Fig. 2 shows the text processing steps that each document goes through. The first step is Tokenization; this operator splits the text of a document into a sequence of tokens, each representing a word. As Arabic is a morphologically rich language, the splitting process can be simple using spaces between words. The second step is filtering the document from the unwanted word. The filters used in our experiment are as follows:

- Arabic Filter Stopwords: removes common words, which have no effect or difference in meaning on the detection of the polarity of a sentence or document [14] like "من"(from), "على" (on), etc. Fortunately, RapidMiner comes with a custom filter for Arabic stopwords which we are going to use.
- Filter Stopwords (dictionary): removes all words equal to a stopword from the given file, an extended list of available stopwords that are available at [15], it contains 750 Arabic stop words.
- Filter Tokin by length: ignored tokens based on their length (i.e. the number of characters they contain), Useless words < 2 characters and >25 characters are removed.

The stemming is a method of word standardization used to match some morphologically related words. The stemming algorithm is a computational process that gathers all words that share the same stem and have some semantic relation, the main objective of the stemming process is to remove all possible affixes and thus reduce the word to its stem. Most Arabic language stemming approaches fall into two classes: root based stemming and light stemming. In our experiment, we compare them with different machine learning classifiers.

3. Representing Documents

To analyze a huge collection of documents, the documents must be transformed from unstructured texts document into structured data as numerical vectors which can be handled by data mining techniques and should be applied for both training and testing documents. The most popular approach for document representation is the vector space model (VSM) [16], Fig. 3 depicts how each document is expressing as t-dimensional vectors $Doc_j = (w_{1j}, w_{2j}, \ldots, w_{ij})$ then building the model.

- **Weighting:** There are several ways to define the weight w_{ij} of a term t_i in a document d_j:
    1. Term Occurrences (TO): A numeric count of the word (W) in the document.
    2. Binary Term Occurrences (BTO): A binary representation indicating the existence of the word, 1 if it exists; 0 otherwise.
    3. Term Frequency (TF): A ratio representing the number of word occurrences over the total number of words.
    4. Term Frequency-Inverse Document Frequency (TFIDF): A numerical representation that is the multiplication of Term Frequency Inverse Document Frequency. It is used for determining the importance of words/phrases or sentences within a given text, against the corpus.

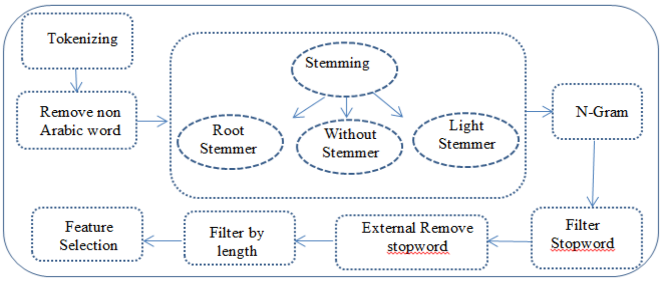

Fig. 2:Preprocessing steps

4. Feature Selection on Movie Review

Actually, in sentiment analysis for movie review which contains large features or words, not all words are necessary for determining the class polarity of the review. Most of the words are irrelevant to the class label. To extract features or words from a large dataset, will most likely produce a large vector space model (VSM). This employs two difficulties. First, because a great number of features can be extracted from these reviews to be used in the model design, the classifier learning process will then have a high computation cost, making it longer to learn the classification function. Second, using a large and sparse feature space might induce the model to overfit the data, and thus present a weak performance when classifying new input information. On the other hand, a good feature or word for classification is the one that has maximum relevance with the output class [11].

Feature selection is defined as the task of eliminating less relevant or redundant features from a given set so feature selection for a word in sentiment analysis is a crucial part, in this paper, we proposed an information gain based feature selection with a threshold. In addition, we also proposed classification schemes based on the Machine learning that is constructed by selected features.

4.1 Information Gain

In the Sentiment Analysis domain, information gain is used to measure the relevance of attribute in class, the higher the value of mutual information between classes and attribute, the higher the relevance between them.

$$IG(C, A) = (C) - H\,(C \mid A) \qquad (1)$$

Where $(C) = -\Sigma cEC\ (C)\ \log\ (C)$, the entropy of the class, and $(C \mid A)$ is the conditional entropy of class given attribute, $(C \mid A) = -\Sigma cEC\ (C \mid A)\ \log\ (C \mid A)$. Since the Cornell movie review dataset has a balanced class, the probability of class C for both positive and negative is equal to 0.5. As a result, the entropy of classes (C) is equal to 1. Then the information gain can be formulated as

$$I\ (C,) = 1 - (C \mid A) \qquad (2)$$

The minimum value of (C, A) occurs if only if $(C \mid A) = 1$ which means attribute A and classes C are not related at all. On the contrary, we tend to choose attribute A that mostly appears in one class C either positive or negative. In the other words, the best features are the set of attributes that only appear in one class. It means the maximum $(C \mid A)$ is reached when (A) is equal to $(A \mid C1)$ documents resulting in $(C1 \mid A)$ and $(C1 \mid A)$ being equal to 0.5. When $(A) = (A \mid C1)$, then the value of $(A \mid C2)$ results in $(C2 \mid A) = 0$ and $(C1 \mid A) = 0$. The value of (C, A) is varied from 0 to 0.5 [17].

4.2 Term Pruning

Pruning, in machine learning, refers to an action of removing less relevant features from the vector space model (VSM).In sentiment analysis; pruning is a useful pre-processing concept because most words in the text corpus are low-frequency words. Perceptual Pruning Ignore words that appear in less than percent of all documents and more than the percentage of all documents. For term pruning, we used the perceptual method with a minimum threshold of 3% and a maximum of 30%.

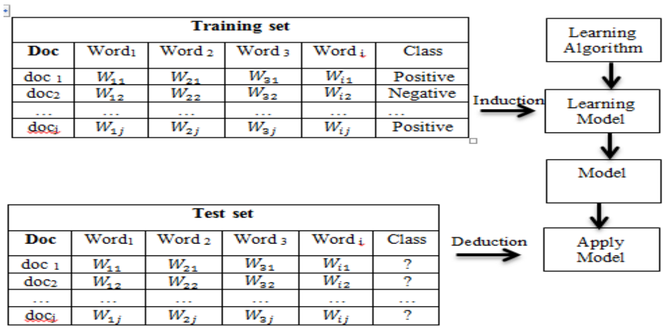

Fig. 3: General Approach for building a classification model for Arabic Sentiment Analysis

5. Classification algorithms

Data Classification is consists of two steps process: (1) the training (or learning) phase to build a set of models and (2) the test (or evaluation) phase where the actual class of the instance is compared with the predicted class as shown in Fig. 3. If the hit rate is acceptable to the analyst, the classifier is accepted as being capable of classifying future instances with unknown class.

The goal of Machine Learning Algorithms is to build a set of models that can correctly predict the polarity of the given document. In our experiments, we used three well known supervised learning methods which are: NB, KNN, and SVM.

A Naive Bayes Classifier (**NB**) is a probabilistic classifier based on Bayes' theorem [18], Bayes' theorem assumes that all words in the corpus are conditionally independent(assume that the effect of an attribute value on a given class is independent of the values of the other attributes), and calculates the probability of them by computing the frequency of features and the relationship between them in the corpus, NB works well in practice even when the dimensionality of the input is high [19][20]. The NB classifier computes a posteriori probabilities of classes, using estimates obtained from a training set of labeled documents.

$$\mathbf{P(class\ |document)} = \frac{p(class).p(documnent\ |class)}{p(document)} \qquad \mathbf{(3)}$$

K-Nearest Neighbor (K-NN) is a method to classify unknown documents. In the training phase, documents have to be indexed and converted to a vector representation. To classify new documents; the similarly of its document vector to each document vector in the training set has to be computed. The only learning task in K-nearest neighbor classifiers is to select two important parameters: the number of neighbors K and distance metric [21] [22], for example, the Euclidean distance, cosine similarity, Jaccard similarity and etc. , KNN was employed with K=15 and Jaccard similarity function which is the best from other distance metric according to [23].

Support Vector Machine (SVM) is a supervised learning method based on the hyperplane that separates between documents vectors in one class from documents vectors in other classes [24]. Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, to map the original data into high dimensional spaces, ANOVA kernels used in our experiments as a good kernel to obtain high accuracy.

The use of the text weighting has a different impact on NB, K-NN and SVM algorithms, In order to apply the empirical evaluation of the Arabic Sentiment Analysis; the OCA Corpus needs to be handled with different settings. The data preparation involves applying the word weighting methods: FT-IDF, to the word N-grams (Bigram). We used UTF-8 format to read Arabic documents then the dataset is divided into two subsets. A training subset is used to build the classification models and a testing subset is used to test the accuracy of the classifier.

In the RapidMiner tool, the Pruning perceptual method was used with a minimum of 3% and a maximum of 30%. It ignores words less than or higher than the minimum and maximum Pruning perceptual threshold in all documents.

IV. EVALUATION PARAMETER

The parameters helpful to evaluate the performance of a supervised machine learning algorithm is based on the element from a matrix known as confusion matrix as shown in Table 2.

A confusion matrix contains information about actual and predicted classifications that are done by a classification system. The performance of such systems is commonly evaluated using the data in the matrix.

TABLE 2 : Confusion Matrix

	Correct Label	
	Positive	**Negative**
Pred. Positive	TP (True positive)	FP (False positive)
Pred. Negative	FN (False negative)	TN (True negative)

1. **Accuracy**(ACC)

The accuracy (ACC) is the proportion of the total number of predictions that were correct.

$$\textbf{Acc} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \textbf{(4)}$$

2. **Precision(Pre)**

$$\textbf{Pre} = \frac{TP}{TP+FP} \qquad \textbf{(5)}$$

i.e. (Correctly retrieved documents) / (Total number of document retrieved)

3. **Recall(Rec)**

$$\textbf{Rec} = \frac{TP}{TP+FN} \qquad \textbf{(6)}$$

i.e. (Correctly retrieved documents)/ (Total number of relevant document retrieved).

V. EXPERIMENTAL RESULT

In this section, we presented and analyzed experimental results of Classification algorithms Naïve Bayes (NB), K-Nearest Neighbors (KNN) where K=15 and distance measure is Jaccard similarity and Support Vector Machines (SVMs). For testing and validation purposes, we applied the 15-fold cross-validation technique instead of 10-fold. We used the RapidMiner software to carry out the experimentation. This tool is built for data mining and machine learning purposes and contains a built-in text pre-processing tasks such as tokenization, stop words removal, weighting schemes, stemming, etc, All classifiers were run in on core i7, 64-bit machine with 8GB RAM. We compared algorithms with and without Information Gain IG as a feature selection method.

Experimental results investigate classifiers' accuracy, the impact of Root stemmer and light stemmer analysis; n-gram models (Bigram) with cross-validation method (k=15).

This paper has addressed sentiment analysis in Arabic review. A dataset, which consists of 500 reviews, The NB, K-NN and SVM classifiers were used to determine the polarity of the review. These classifiers built their classification models by using two versions of the same dataset. The first experiment is using the feature selection technique in the preprocessing steps and the second experiment without using the feature selection technique.

TABLE 3:performance **NB** classifier with using feature selection

	NB		
	ACC	**Pre**	**Rec**
Root Stemmer	89.98%	95.53%	83.92%
light stemmer	90.40%	95.87%	84.80%
Without stemmer	89.60%	95.36%	83.60%

TABLE 4:performance **KNN** classifier with using feature selection

	KNN		
	ACC	**Pre**	**Rec**
Root Stemmer	92.18%	93.80%	90.39%
light stemmer	90.97%	93.21%	88.43%
Without stemmer	90.97%	93.80%	87.99%

TABLE 5:performance **SVM** classifier with using feature selection

	SVM		
	ACC	**Pre**	**Rec**
Root Stemmer	92.18%	93.80%	90.39%
light stemmer	90.97%	93.21%	88.43%
Without stemmer	90.97%	93.80%	87.99%

Tables 3-5 illustrate performance NB, KNN and SVM classifiers with TFIDF weighting with implement Information Gain IG method as a feature selection method in the pre-processing step, and SVM achieves the best accuracy with Root stemmer. NB and KNN are achieved high accuracy without using the stemmer. From the result, stemming is not always beneficial for text categorization as shown in Fig. 4.

TABLE 6: performance **NB** classifier without using feature selection

	NB		
	ACC	**Pre**	**Rec**
Root Stemmer	87.37%	93.96%	79.95%
light stemmer	88.79%	93.36%	84.00%
Without stemmer	89.20%	95.31%	82.82%

TABLE 7:performance **KNN** classifier without using feature selection

	KNN		
	ACC	**Pre**	**Rec**
Root Stemmer	87.59%	90.90%	83.58%
light stemmer	85.40%	92.15%	77.67%
Without stemmer	89.80%	93.97%	85.17%

TABLE 8:performance **SVM** classifier without using feature selection

	SVM		
	ACC	**Pre**	**Rec**
Root Stemmer	90.80%	92.30%	89.19%
light stemmer	89.78%	90.38%	89.19%
Without stemmer	90.56%	92.30%	88.75%

Tables 6-8 illustrate performance NB, KNN and SVM classifiers with TFIDF weighting without implement Information Gain IG method. Information Gain IG method provides the classifiers fast, and more accurate.
The feature selection IG improves the accuracy of NB, KNN, and SVM as shown in Fig. 4.

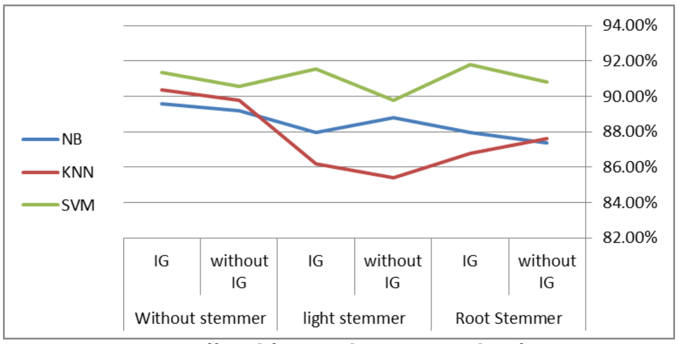

Fig. 4: affected feature selection IG on classifier

From Fig. 4 the SVM classifier achieves the best performance than other classifiers and produces height accuracy of 92.38% with Root Stemmer and feature section. KNN classifier gets Worst accuracy about 85.40% with light stemmer without feature selection but increased to 90.97% when using the feature selection.

TABLE 9 comparisons the accuracy

Classifier	**With FS**	**Without FS**	**Improvement**
NB	89.98%	87.37%	2.61%
KNN	92.18%	87.59%	4.59%
SVM	92.38%	90.80%	1.58%

Table 9 shows the effect of feature selection IG and threshold in the accuracy of NB, KNN and SVM classifiers.
With using feature selection IG and threshold, NB classifier increased about 2.61%, KNN classifier increased about 4.59%and SVM classifier increased by about 1.58 %.

VI. CONCLUSIONS

This work has considered sentiment analysis in Arabic text**,** The NB, SVM and KNN classifiers were used to detect the polarity of a given review into positive or negative class with respect to their contents. RapidMiner machine learning tool is used to build the classifier that includes across validation process (k=15) to estimate the performance of the classifier. Feature selection algorithm (IG) with threshold has a great influence on the accuracy of text categorization reach to 2.6% for BN classifier, 4.5%.for the K-NN classifier and 1.58% for SVM classifier.
By comparing the results**,** the use of the text weighting has a different impact on algorithms. The KNN and SVM machine learning technique achieves the highest accuracy when the Light Stemmer is used with feature selection. Whereas, without using feature selection and stemmer NB and KNN machine learning technique achieves the highest accuracy.

REFERENCES

[1] B. Brahimi, M. Touahria and A. Tari, "Data and Text Mining Techniques for Classifying Arabic Tweet Polarity," *Journal of Digital Information Management,* pp. 15-25, 2016.

[2] R. M. Duwairi, "Sentiment Analysis for Dialectical Arabic," *International Conference on Information and Communication Systems,* 2015.

[3] A. Tripat, A. Agrawal and S., K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems With Applications,* pp. 117-126, 2016.

[4] Abdulla, Nawaf A., Mahmoud Al-Ayyoub, and Mohammed Naji Al-Kabi, "An extended analytical study of Arabic sentiments," *International Journal of Big Data Intelligence 1,* vol. 1, pp. 103-113, 2014.

[5] S. Atia and K. Shaalan, "Increasing the Accuracy of Opinion Mining in Arabic," *2015 First International Conference on Arabic Computational Linguistics,* no. IEEE, pp. 106-113, 2015.

[6] Mohammed Rushdi-Saleh, M. Teresa Martín-Valdivia, L. Alfonso Ureña-López,, "OCA: Opinion Corpus for Arabic," *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY, 62(10):2045–2054,* 2011.

[7] D. A. Said, N. M. Wanas, N. M. Darwish and N. H. Hegazy, "A Study of Text Preprocessing Tools for Arabic," in *In The second international conference on Arabic language(pp. 230-236),* JANUARY 2009.

[8] Motaz K. Saad, Wesam Ashour, "Arabic Text Classification Using Decision Trees," in *In Proceedings of the 12th international workshop on computer science and information technologies CSIT,* Moscow – Saint-Petersburg, Russia, 2010.

[9] Rehab Duwairi, Mahmoud El-Orfali, "A Study of the Effects of Preprocessing Strategies on Sentiment Analysis for Arabic Text," *Journal of Information Science,* pp. pp. 1-14, 2013.

[10] Rehab M. Duwairi, Mohammad Nayef Al-Refai, Natheer Khasawneh, "Feature Reduction Techniques for Arabic Text," *Journal of the American Society for Information Science and Technology,* p. 60(11):2347–2352, November 2009.

[11] M. Rogati and Y. Yang, "High-Performing Feature Selection for Text Classification," *Proceedings of the eleventh international conference on Information and knowledge management,* pp. 659-661, 2002.

[12] A. I. Pratiwi, "On the Feature Selection and Classification Based on Information Gain for Document Sentiment Analysis," *Applied Computational Intelligence and Soft Computing,* vol. 2018, pp. 1-5, 2018.

[13] "Sistemas Inteligentes de Acceso a la Información," [Online]. Available: http://sinai.ujaen.es/es/?s=Corpus&submit=Search.

[14] R. Nalawade, A. Samal and K. Avhad, "Improved Similarity Measure For Text Classification And Clustering," *International Research Journal of Engineering and Technology (IRJET),* vol. 03, no. 05, pp. 214 - 219, 2016.

[15] "Github," [Online]. Available: https://github.com/mohataher/arabic-stop-words/blob/master/list.txt. [Accessed 2018].

[16] R. Al-Shalabi, G. Kanaan and M. H. Gharaibeh, "Arabic Text Categorization Using kNN Algorithm," in *The 4th International Multiconference on Computer and Information Technology, CSIT,* 2006, April.

[17] Asriyanti Indah Pratiwi and Adiwijaya," On the Feature Selection and Classification Based on Information Gain for Document Sentiment Analysis",5 pages, 2018.

[18] J. Han, M. Kamber, and J. Pei, "Data mining: concepts and techniques," in *Naive Bayesian Classification,* Morgan Kaufmann, 2012, pp. 351-355.

[19] R. Raghuwanshi, "A Comparative Study of Classification Techniques," *International Journal of Computer Science and Information Technologies,* pp. 78-82, 2016.

[20] Nawaf A. Abdulla, Nizar A. Ahmed, Mohammed A. Shehab, and Mahmoud Al-Ayyoub, "Arabic Sentiment Analysis: Lexicon-based and Corpus-based," in *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT),* JANUARY 2013.

[21] w. Hämäläinen and M. Vinni, "Classifiers for educational data mining," *Handbook of educational data mining,* pp. 57-74, 2010.

[22] J. Wang, P. Neskovic and L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure," *Pattern Recognition Letters,* pp. 207-213, 2007.

[23] Hana. A. Al-Hatibi and . A. S. M. Haider, "Similarity-based Learning Approaches to Arabic Sentiment Analysis," *Journal of Environmental Science, Computer Science and Engineering & Technology,* vol. 8, pp. 58-70, 2019.

[24] R. M. Duwairi, "Arabic Sentiment Analysis Using Supervised Classification," in *Future Internet of Things and Cloud (FiCloud),2014 International Conference on IEEE,* 2014.