JIS

Article

A Study of the Effects of Preprocessing Strategies on Sentiment Analysis for Arabic Text

Journal of Information Science
1–14

DOI: 10.1177/0165551510000000
jis.sagepub.com
SAGE

Rehab Duwairi
Department of Computer Information Systems, Jordan University of Science and Technology, Jordan

Mahmoud El-Orfali
Department of Computer Science and Engineering, Qatar University, Qatar

Abstract
Sentiment analysis has drawn a considerable interest among researchers due to the realization of its fascinating commercial and business benefits. This paper deals with sentiments analysis in Arabic text from three perspectives. Firstly, several alternatives of text representation were investigated. In particular, the effects of stemming, feature correlation, and n-gram models for Arabic text on sentiment analysis were investigated. Secondly, the behavior of three classifiers, namely, SVM, Naïve Bayes, and K-nearest neighbor classifiers, with sentiment analysis was investigated. Thirdly, the effects of the characteristics of the dataset on sentiment analysis were analyzed. To this end, we have applied the techniques proposed in this paper to two datasets; one was prepared in-house by the authors and the second one is freely available online. All the experimentation was done using Rapidminer[1]. The results show that our selection of preprocessing strategies on the reviews increases the performance of the classifiers.

Keywords
Sentiment Analysis, Opinion Mining, Arabic Text, Polarity Classification.

I. Introduction

In the last decade, text categorization or classification has dominated a considerable interest among researchers and developers in the field of data mining due to the need to extract and organize valuable information effectively and automatically for business intelligence applications and users. Many techniques have been introduced to solve this problem, in which the main focus is on topic categorization, where the documents are being classified into classes according to their subject matter [1, 2]. One recent classification technique, however, is classifying the documents based on users' opinions towards the subject matter or more specifically, classifying them according to their sentiments [3, 4, 5, 6]. For example, rather than categorizing some documents as financial documents, these documents can be classified further as positive or negative reviews on a certain financial data [7].

This field of science has emerged as a result of the exponential growth of many social web sites, blogs and forums that are rich with people opinions on several diverse fields that include different products, books, movies, articles, and many others which made the sentiments widespread. With the fact that knowing sentiments towards services was always an important piece of information that companies need to know, sentiment analysis has become a demand and it is increasingly employed in many commercial and business intelligence applications. It is possible now for companies to have certain means to estimate the extent of their product acceptance which would help in determining strategies to improve the products quality.

Sentiment analysis is a very difficult task as people have many ways of expressing their opinions. For example, one would express an opinion as "This is a great product" which is a positive opinion while another would say "One might think that this is a great product". In this sentence, the phrase "might think" has changed the whole meaning of the expressed opinion. Also, there is a great deal of difficulty in differentiating between facts and opinions in a text and to

Corresponding author:
Rehab Duwairi, Department of Computer Information Systems
P.O.Box. 3030, Jordan University of Science and Technology, Irbid 22110, Jordan
Rehab@just.edu.jo

identify subjectivity hidden between the lines. All of this encouraged the researchers to try different techniques in which some were focusing on the preprocessing part and others were concentrating on the classification algorithms.

The majority of work on sentiment analysis has targeted English text whereas a language like the Arabic language which holds the fifth position among the most 30 languages spoken worldwide [8] did not get enough focus. Also, to the best of our knowledge, there is no Arabic lexical resource available for opinion research purposes other than the one prepared by Saleh, Martín-Valdivia, Ureña-López and Perea-Ortega [9] which falls under the movies domain.

This paper presents a new dataset, which consists of reviews written in Arabic, to be used in sentiment analysis. It also presents several preprocessing and feature representation strategies and assesses their effects on the accuracy of sentiment analysis. The behavior of SVM, Naive Bayes and K-NN classifiers, with respect to different preprocessing techniques and datasets, was recorded. To make our work directly comparable to other existing work, we experimented with a second dataset that is publically available. All the experiments were carried out using Rapidminer (http://rapid-i.com).

The rest of this paper is organized as follows. Section 2 provides some background on sentiment analysis. Section 3, on the other hand, presents related work. Section 4, by comparison, introduces our approach and general design. Section 5, explains implementation strategies. Section 6 discusses experimentations and results. Finally, section 7 presents the conclusions of this work.

2. Background

2.1. Sentiment Analysis

Sentiment analysis is an interdisciplinary subject that utilizes techniques from natural language processing (NLP), text mining and computational linguistics to identify and extract subjective information from source materials [10]. In other words, it is the function that tries to analyze emotions, opinions and reviews to classify them into one of two polarities: positive or negative. In fact, opinions are the main drivers of our behaviors. People's beliefs and perceptions of real life matters are in most cases dependent on how others see the world. Opinion mining as a concept is not new. In the past, when people need to make important decisions, they usually ask their friends for their opinions. In companies, getting customers' feedback is done using customer surveys, blogs and suggestion committees. This showed that opinions by themselves are considered as a very valuable source of information. Sentiment can be expressed at four different levels which are heavily interconnected:

- *Document Sentiment Classification*: Probably the most widely studied problem [7]. Here, the attempt is to try to classify a whole a review, as either positive or negative. In some cases neutral class is also considered. It is very similar to a topic-based text classification in the sense that the whole review is treated as one unit and is mapped to one feature vector. The whole review is assigned to the positive, negative or neutral classes. Of course, one has to pay attention to the properties of sentiment analysis, for example, dealing with negation words, such as not, which could flip the direction of the sentiment from positive to negative or vice versa.
- *Sentence Subjectivity & Sentiment Classification*: Here, the concern is to identify subjective sentences from objective ones and classify these sentences into positive or negative sentences. This is probably useful and true for simple sentences but it is not the case in comparative sentences. e.g. "I like the book but it is too long".
- *Aspect-based Sentiment Analysis*: Sentiment analysis at both document and sentence levels are useful to some extent but they lack the ability of finding what people liked and disliked. That is, they don't find the targets of the extracted opinions. This makes the use of opinions limited. Targets in opinions are the entities and their features. I-phone is an entity but battery-life is a feature.
- *Aspect-based Opinion Summarization*: This level actually considers all the opinions on all the aspects in different documents and tries to summarize the positive or negative feedback. For example it tries to infer and extract information in percentages: "70% of people liked i-phone".

Sentiment analysis has many challenges. This has several reasons; one reason is that people have different writing styles. The same word can be considered positive in one situation and negative in another. For example, the word "long" is considered as a positive opinion in the sentence "The laptop battery's life is long" but it is considered a negative opinion in the sentence "The laptop boot time is long". Also people opinions change over time. A much bigger

challenge in sentiment analysis comes from the fact that people usually express their opinions in a comparative manner; and, people tend to express their positive and negative reviews in the same sentence.

2.2. Arabic Language

Arabic language differs from English, German and other languages in its orthographical nature. There are 28 Arabic letters. The words are written from right to left and the shape of these letters changes according to their position in the word: beginning, middle, end, connected to the previous letter or not connected. For example, the letter b "ب" can assume several shapes: " بـ " at the beginning of words as in "باب " which means door; " بـ " in the middle of words as in " مكتبة " which means library; " ب " at the end of words when connected to the previous letter as in " كتب " which means books; " ب " at the end of words not connected to the previous letter as in " كتاب " which means book. Although machine learning does not require deep linguistic knowledge but Arabic has its challenges. The simple tokenization step, which is common in machine learning, is not easy for Arabic as Arabic neither supports letter Capitalization nor has strict punctuation rules. For English, this is easy, the sentence starts with a capital letter and ends with a period. Arabic is a morphologically rich language and this is also complicates tokenization; one Arabic word could contain four tokens.

Stemming, which is a very important task in NLP, is another challenge in Arabic. In non-Arabic languages, a basic stem can either be pre-fixed or post-fixed to express a grammatical syntax. In Arabic, on the other hand, there is, in addition to these affixes, the infix where the stem additions can be within the root itself. This adds a real problem when applying stemming to Arabic documents as it became harder to differentiate between root letters and affix letters. Generally, there are two approaches to stemming. The first approach reduces a word to its three-letter root. The second approach, called light-stemming, removes common suffixes and prefixes without reducing a word to its root. Stemming is not accurate for Arabic due to the fact that most stemmers focus on reducing words to three-letter roots while some words have four-letter or five-letter roots.

In addition to the above, Arabic has three varieties: Classical Arabic (Found is religious scripts), Modern Standard Arabic or MSA, found in today's written Arabic and spoken in formal channels and Colloquial or dialectical Arabic – the spoken language in informal channels. Arabic dialects vary from one Arab country to another. When dealing with reviews published in social media channels, colloquial Arabic is heavily present. In this work, both stemming and light stemming were used as preprocessing strategies as shown in Section 4.

3. Related Work

This section presents previous work that deals with sentiment analysis. The below listed works vary in their preprocessing strategies, analysis methods, and structure of reviews. Some used supervised learning, others used semi-supervised learning. They also differ in the language of the reviews. The work of Pang, Lee and Vaithyanathan [11] showed that importing the algorithms and techniques used for text categorization does not necessarily yields good results. This is understandable as sentiment analysis is different from text categorization in many aspects. For example, in text categorization, negation words are considered stopwords and thus removed while in sentiment analysis these are considered important words and thus retained. The preprocessing techniques that were employed in this work varied from using bag-of-word model to part-of-speech (POS) tagging and different n-gram models. In this work, we have extended these preprocessing techniques as explained in Section 4.

Bickerstaffe and Zukerman [12] adopted a hierarchical multi-classifier model to infer the sentiment of a review on a discrete scale; stars system. The aim was to enhance the results obtained by [20] and the authors concluded that their approach was competitive and in some cases outperformed most of the classifiers presented in [11]. The current proposed framework handles unstructured text to extract sentiment which is more challenging.

Paltoglou and Thelwall [13] employ sophisticated variations of the standard *tf-idf* feature weighting to their SVM classifier in comparison to the binary weighting scheme. They expressed the importance of using a term weighting function that scales sub-linearly according to the term frequencies in the document provided that a document frequency smoothing factor is adapted. They concluded that their work has added a significant increase in the accuracy compared to the other state-of-the-art approaches. The preprocessing that was done here is related to the term weighing scheme.

Hall [14] introduced a correlation-based feature selection technique for machine learning and incorporated that into sentiment analysis. The idea of the work is to keep the highest correlated features to the polarity class and remove the least correlated ones. In this approach, the feature-class correlation has to be high in contrast to the feature-feature correlation which has to be low. The *merit* is computed of each set combinations to determine the best subset of features

that are the highest correlated features to their class. Different variations of the correlation function are used depending on the types of feature - whether they are numeric, nominal or mixed features. This approach has showed that feature selection based on feature-class correlation yields a remarkable improvement in the accuracy. The above three works used supervised learning for sentiment classification. They differ in their respective preprocessing strategies and their employed classifiers.

Esuli and Sebastiani [15] tried to determine the sentiment orientation of a subjective sentence by the use of the glosses of terms in a semi-supervised classification. This resulted in different term representations as the glosses are generated from the terms' definitions given in online dictionaries. This work is close to lexicon-based sentiment analysis where the polarity of words is determined a priori irrespective of the context.

Sarvabhotla, Pingali and Varma [16] tried to extract subjectivity by minimizing the dependency on the resource lexicons by applying a statistical methodology called review summary used in combination with a mutual information feature selection method. A number of researchers relied on lexical resources such as dictionaries and thesauri to extract sentiment terms from reviews in order to determine their polarity. Denecke [17], Ohana and Tierney [18] have followed this approach by the use of SentiWord [19] which is an open lexical resource dedicated for sentiment analysis.

Most of the work on sentiment analysis has targeted the English language. However, Denecke [20] has extended his work in [17] to use the SentiWordNet [19] for multilingual sentiment analysis. In this study, German movie reviews were subjected to language identification process in which it will be translated to English. Once this is done, NLP processing is applied to remove stopwords and to apply stemming on the documents' words. After that, the scoring algorithms used before are applied with some machine learning algorithms. Although this approach would employ some errors in the translation as it is done as a standard approach, such methodology can be considered as a viable approach within the multi-lingual approach. Kosinov's [21] represented the reviews as character n-grams.

El-Halees's work [22] is one of the few papers that deal with Arabic sentiment analysis. A combined classification approach which consists of three methods (lexicon-based classification, ME and K-NN) is used to predict the document polarity. The work is done by applying these three classifiers in sequence. The lexical-based classifier is used to determine the opinion terms and their sentiments and is built by preparing a dictionary of positive and negative words from two sources. The first one is an online dictionary and the second one is from SentiStrength [23] which has a dictionary of 2600 scored English sentiment words. These words have been translated to Arabic language with their same scoring strengths as an attempt to overcome the lack of having a solid Arabic sentiment lexicon. Once this is done, a maximum entropy classifier (ME) is applied in such a way that it will classify the documents having their sentiment probability greater than a certain threshold so that the total number of documents will be refined before feeding them into the third step which is training with K-NN. This is a hybrid approach which utilized sentiment lexicons in addition to supervised learning. Their dataset is not publically available and therefore we cannot do a direct comparison between our work and theirs.

Ahmad and Almas [24] aimed at extracting sentiments from financial texts. The motivation of the work is to find some grammars or rules describing common and frequent Arabic patterns that are usually used in financial news to report object values' changes like the change in shares value. Also, movements, hedges, quantities and measure patterns are also considered. These rules are used then to visualize the changes in sentiments embedded in the news text. They have extended their work with Cheng and used the same idea of building a language idiom in [25]. They developed a local grammar method for three language idioms (Arabic, Chinese and English) in which these idioms have been created by comparing a distribution of domain-specific words with a distribution of general-corpus words. The work in [24] is closer to lexicon-based learning where the goal was to detect grammars that are frequently used to convey positive sentiments and grammars that mostly convey negative sentiments.

Another study that focused on multi-language sentiment classification is done by Abbasi, Chen and Salem in [26]. They concluded that using a feature selection method as a preprocessing strategy would affect positively the accuracy of analysis.

Elhawary and Elfeky [27] have also worked in the domain of financial news where they have developed an engine that crawls the web for Arabic financial reviews and determine their sentiments. They used a list of more than 600 positive phrases and 900 negative phrases to build an Arabic Similarity Graph which is used as a comparison mechanism to know the review sentiments. Evaluation showed that the Arabic sentiment classifier has approximately a similar performance of any English sentiment analysis approach.

Finally, Saleh, Martín-Valdivia, Ureña-López and Perea-Ortega [28] have built a corpus that consists of reviews about movies. 250 positive reviews and 250 negative reviews collected from different web pages. They applied different NLP tasks on these reviews including stopword elimination, stemming and n-grams generation for unigrams, bigrams

and trigrams. SVM and Naïve Bayes are used to classify the reviews. Their preprocessing techniques are close to what we have used here. However, we have employed a larger set of preprocessing techniques as it is explained in Section 4.

As it can be seen from the above description of the selected literature review, preprocessing is an integral part of any framework that handles text. Preprocessing strategies cover a wide spectrum of choices. It starts by eliminating punctuation marks and foreign symbols from the text. Second, stopwords are removed but this is tricky for sentiment analysis as valence shifters, such as negation, play a crucial role when determining the polarity of a text. Third, stemming can be used to reduce the number of words in the vocabulary. Fourth, weights of remaining terms are calculated using one or combination of term frequency, tf-idf and so on. Fifth, feature selection or reduction techniques can be used to reduce the number of words in the vocabulary, or to enhance the results. The purpose of this paper is to provide an insight on the effects of several preprocessing strategies on sentiment analysis' accuracy when applied to Arabic text. Arabic is a morphologically rich language which needs special care during the preprocessing stage. The above related work which deals with sentiment analysis in Arabic text did not provide a comprehensive framework for preprocessing strategies contrary to this work.

4. Approach and General Design

This section describes the framework that we have used to extract sentiment from Arabic reviews. In general, there are three main approaches that one can handle this problem, namely, using supervised learning, using lexicons or using hybrid of both. Supervised learning means that a classifier or a set of classifiers is trained to predict the sentiment of new unseen reviewers by building a classification model from labelled reviews. This means that there must exist a labelled set of reviews for the supervised techniques to work. Reviews labelling or annotation is a human-based task that requires huge efforts. Recently, crowdsourcing was used to get labels from anonymous individuals. The second approach, on the other hand, relies on a lexicon or dictionary of words and their respective sentiments. Every word in the lexicon is labelled as being positive, negative or neutral and to what degree. The idea of lexicons is that words, regardless of their context, carry sentiment. For example the word "love" indicates positive sentiment and the word "hate" indicates negative sentiments. Of course one can argue against the validity of this hypothesis, however, there are a good number of researchers who have employed lexicons and obtained good results. The main advantage of using lexicons is that there is no need to have a labelled dataset. The Hybrid approach tries to augment classifiers with lexicons to enhance their accuracy. Evidence, from the literature, shows that supervised learning performs better than lexicons for domain specific problems.

In the current research we have relied on supervised learning to extract sentiment in reviews. This means that we needed an annotated dataset. To this end, we have developed our own dataset in addition a publically available dataset. With supervised learning, it is custom to represent the text or reviews as a bag of words. This means that every review is tokenized into words or tokens. The importance or weight of a word is usually determined using tf-idf as shown in Equation (1). The order of words is not important [29].

In our work, we have three research questions that we wanted to find answers to. The first question deals with reviews' representation: is the simple bag-of-words model good for sentiment analysis? Will the accuracy improve if this simple model augmented with say n-grams? The second research question deals with the classifier that is used, will all classifiers give the same performance? Are there classifiers that are most suited to sentiment analysis than others? The third question deals with the dataset, does the accuracy of sentiment analysis frameworks give different results when applied to different datasets? The next three points describe our input to answer each of the above questions. The current work deals with Arabic sentiment analysis from three perspectives:

(1) *Representation perspective*: the reviews were represented in seven different models as a pre-classification task. The purpose of these alternative representations is to assess their effects on accuracy. The seven representations are: base-line vectors that are built using the actual words in the reviews without any processing; stemmed vectors that apply stemming to words before calculating tf-idf; vectors which include the remaining words after applying feature reduction; vectors that are built using word-n-grams; vectors that were created using character-n-grams; vectors that are built using word n-grams after applying feature correlation reduction; and finally vectors that were created using character n-grams with feature correlation. Table 1 shows how these seven representations were used in sentiment classification.

(2) *Classifier perspective*: Three supervised learning approaches (SVM, K-NN and Naïve Bayes) were used to classify the Arabic reviews into negative or positive classes. These three classifiers are considered the most well performed methods in data mining [30]. The behavior of each classifier is studied with respect to the

dataset and to the vector representation model. The idea is to assess which data representation is suitable for each classifier. Also, to assess the classifier sensitivity to the domain of the reviews.

(3) *Dataset perspective*: we analyzed the effects of the data properties on the results obtained. We tried to answer the following questions: how does dialectical Arabic affect sentiment analysis? How does Modern Standard Arabic (MSA) affect sentiment analysis? Do the results depend on the classifiers, on the representation model, or on the dataset? Our findings show that the results somewhat depend on all of the previous three factors.

5. Implementation

5.1. Datasets

The first dataset has been prepared manually by collecting reviewers' opinions from Aljazeera[2] website against different published political articles. The dataset includes 322 reviews and have been evaluated by two evaluators. Only the matching evaluations are kept and any other conflicts in labeling are ignored. Also, neutral reviews have been excluded as this work is not considering the neutral class. This dataset is mostly written by people from the public and consists of short reviews. In total, 164 positive reviews and 136 negative reviews were kept. The number of features for this dataset is 750 features. It includes dialects as well as spelling mistakes. We will analyze the influence of such issues on the classification process in subsequent sections. Table 2 shows the properties of this dataset.

The second dataset represents an Arabic opinion corpus that is freely available for research purposes prepared by Saleh, Martín-Valdivia, Ureña-López and Perea-Ortega in [9, 28]. It has 500 reviews that have been extracted from different movies' web pages; 250 positive reviews and 250 negative reviews. These reviews have been written in Modern Standard Arabic (MSA) by professional reviewers and considered of high quality. The reviews of each dataset were preprocessed in several ways and fed to the three classifiers. 10-fold cross validation was used for both datasets.

Table1: Representation Models for Textual Customer Reviews.

Representation Level	Tasks
Tier 1: baseline vectors	Every review is transformed into a feature vector which includes all the words in that review along with their tf-idf values.
Tier 2: stemming and stopword removal	Here stopwords are eliminated from every review (paying attention to negations as these have effects on polarity classification), then remaining words are stemmed. Stemming is seen as a feature reduction and a way of reducing syntactical variations among reviews. Finally, the tf-idf is calculated for the remaining words.
Tier 3: feature reduction	Feature-class correlations and feature-feature correlations are calculated on the vectors of Tier1 using Pearson coefficient. Only features with high correlation to the class and low correlations to other features are retained. The idea is to keep only independent features in every review.
Tier 4: word n-grams	The feature vectors contain phrases rather than single words. A phrase is an n-gram of words. N-grams help in capturing negated words and impose some order of words. Several values for n were experimented with.
Tier 5: character n-grams	Same as Tier4 but character n-grams rather than word n-grams are used. On the contrary to character n-grams, the phrases in Tier4 are meaningful to the human.
Tier 6:	This is a combination of Tier1,Tier 3 and Tier 4. i.e. feature correlation was applied on word n-grams.
Tier 7:	This is a combination of Tier1,Tier 3 and Tier 5. i.e. feature correlation was applied on character n-grams.

5.2. Vector Generation

The baseline vectors were generated by tokenizing every review. No stemming or stopword removal where applied to the features. The importance of every feature is computed using tf-idf as the following equation shows:

$$Term\ Weight = tf_i * Log(\frac{D}{df_i}) \quad (1)$$

Where:

- tf_i is the term frequency or the number of times the term *i* occurs in a document.
- df_i is the document frequency or the number of documents containing the term *i*.
- *D* is the total number of documents in the corpus.

Table 2: Properties of the Politics Dataset

	Evaluated by Reviewer 1	Evaluated by Reviewer 2	Agreed on
Positive Reviews	167	164	164
Negative Reviews	140	136	136
Neutrals	22	29	22
Total	329	329	322
Total Considered Reviews			300

In Tier 2: stopwords were removed and features were stemmed. Two methods of stemming were used. The first one reduces every feature to its three-letter root (this is called Tier 2a). In the second method, called light stemming, only common prefixes and suffixes were removed (Tier 2b). Rapidminer built-in stemming was used.

Tier 3: Feature-class correlations and feature-feature correlations were calculated using Pearson coefficient. The optimal set of features, which has high correlations with the class and low correlations with other features are used to represent reviews. This set was determined experimentally by starting with one feature and finding accuracy of the classifiers, then new features are added and accuracy is computed again; this process is continued until the addition of new features does not improve the accuracy. Since we are dealing with three classifiers, the optimal set of features is also correlated with the classifier. Table 3 shows the optimal feature sets for the three classifiers for the Movie dataset. For example, the accuracy of the Naïve Bayes classifier reached its maximum when 1200 features were used. For SVM, the number of features was 2100 and for KNN (when K=10), the number is 5000 features.

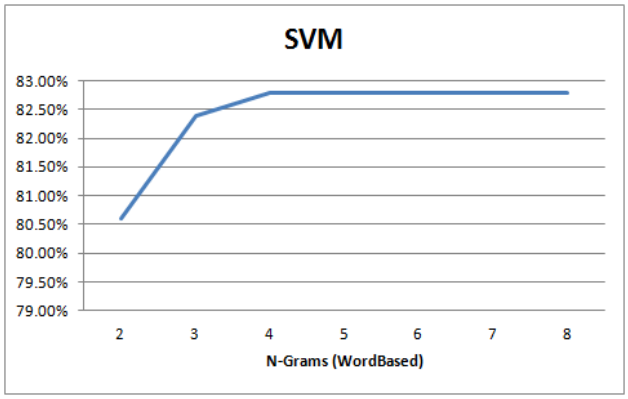
Table 3: Sensitivity Analysis against Number of Features for the Movie Corpus.

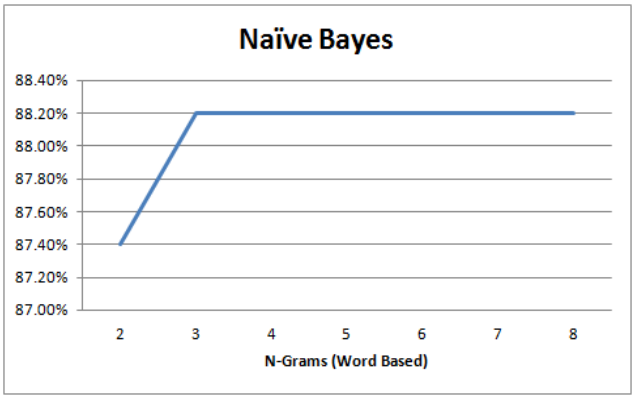
Top Correlated Features	Naïve Bayes	SVM	K-NN
300	92.40%	84.80%	60.20%
600	93.60%	88.00%	58.40%
900	95.80%	87.40%	51.80%
1200	**96.60%**	87.80%	51.20%
1500	96.00%	89.20%	51.00%
1800	95.00%	89.20%	51.00%
2100	94.20%	**89.80%**	52.60%
2500	92.40%	89.60%	50.80%
2800	91.20%	89.00%	50.20%
3200	91.00%	88.40%	51.00%
3500	90.20%	86.60%	51.20%
5000	85.60%	79.60%	**88.00%**

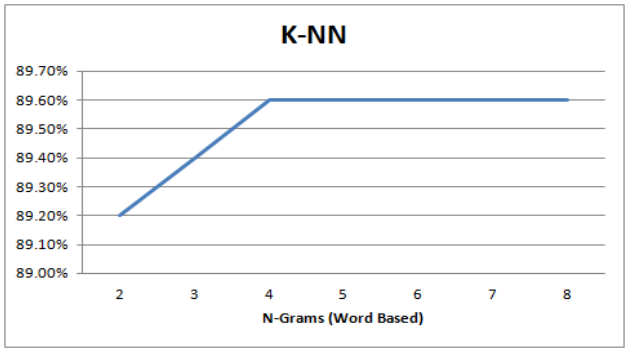
Tier 4 and Tier 5: here the reviews are represented as vectors that consist of word or character n-grams. The value of n was determined experimentally. This means that the best value of n for every dataset and for every classifier was determined by starting with n=1 and calculating accuracy. After that, n was set to two and the accuracy was calculated again and so on. Figure 1 shows the accuracy of the three classifiers for different values of n (in word n-grams) for the Movie dataset (refer to Section 6.1 for the definition of accuracy). Figure 1 shows that SVM and KNN reached their highest accuracy when 4-word-grams were used. By comparison, Naïve Bayes performs best when 3-word-grams were used.

Table 4 shows the accuracy of the classifiers for the Movie dataset. N means the number of characters in the character n-grams. The idea was to determine the best value for N for every classifier when applied to the Movie dataset. The experiment starts by setting N=1 and then calculates the accuracy of the current classifiers, then it sets N=2

and re-calculates the accuracy of the current classifier and so on. The experiment, for a given classifier, stops when the accuracy cannot be further improved.







N-Grams (Word Based)			
N	Naïve Bayes	SVM	K-NN
2	87.40%	80.60%	89.20%
3	88.20%	82.40%	89.40%
4		82.80%	89.60%
5			
6			
7			
8			

Figure 1: Sensitivity Analysis of Classifiers against Different Values of n for Word n-grams.

As Table 4 shows, the three classifiers performed best when N = 7. The hypothesis, which we try to prove here, is that the distribution of character n-grams also affects sentiment analysis. Tier 6 and Tier 7 combine several preprocessing strategies. In Tier 6, after the best value of n is determined, the resultant vectors are processed to determine the set of most highly-correlated n-grams. Tier 7 is very similar to Tier 6, except that character n-grams are used instead of word n-grams. For all tiers, the importance of a feature or a term is expressed using tf-idf as defined previously.

Table 4: Sensitivity Analysis of Character n-grams for the Movie Dataset

N (No. of Characters)	Naïve Bayes	SVM	K-NN
1	62.60%	69.40%	68.00%
2	71.00%	82.80%	77.80%
3	72.40%	77.60%	81.80%
4	80.60%	77.00%	84.40%
5	84.20%	77.40%	85.80%
6	85.00%	88.00%	86.00%
7	**86.00%**	**88.60%**	**88.60%**
8	85.60%	81.80%	88.20%
9	85.40%	88.00%	88.20%
10	85.80%	79.40%	88.20%

6. Experimentation and Results Analysis

6.1. Evaluation Metrics

Accuracy, Precision and Recall were used for evaluating the performance of our suggested framework. Table 5 shows a confusion matrix that will explain how the previous three measures are calculated. For simplicity, the matrix shows a binary classification problem, i.e. we have one class, say C, and both the human and the computer classify n reviews or objects to belong (C) or not to belong to the class (~C).

Table 5: Confusion Matrix

Human	Computer	Category
C	C	tp
~C	C	fp
C	~C	fn
~C	~C	tn

tp (true positives): refers to the number of reviews that both the human and computer agree to belong to the class C. As we have two classes in this work (Positive, Negative) then tp means the number of positive reviews that were correctly classified to belong to the Positive class in addition to the number of negative reviews that were correctly classified to belong to the Negative class.

fp (false positives): the number of reviews that the human classifies them to belong to the class C but mistakenly the classifier categorizes them not to belong to the class C. For our case, this number equals to the number of true positive reviews that were classified as not-Positive and the number of true negative reviews that were classified as not-Negative reviews.

tn (true negatives): the number of reviews that both the human and the computer label them as not belong to the class C. For our case, when dealing with the Positive class, this number equals the number of reviews that do not belong to the Positive class (say V1). When handling the Negative class, this number means the number of reviews that do not belong to the Negative class (Say V2). Then tn=V1+V2.

fn (false negatives): the number of reviews that the human says they belong to the class C but the computer says they do not belong to the class C. For our case, this equals V3+V4. V3 equals the number of positive reviews that mistakenly classified as not to belong to the Positive class. V4= the number of negative reviews that were mistakenly classified as not to belong to the Negative class.

Given the above definitions, the following equations show how to find accuracy, recall and precision:

$$Accuracy = (tp + tn)/(tp + tn + fp + fn) \quad (2)$$

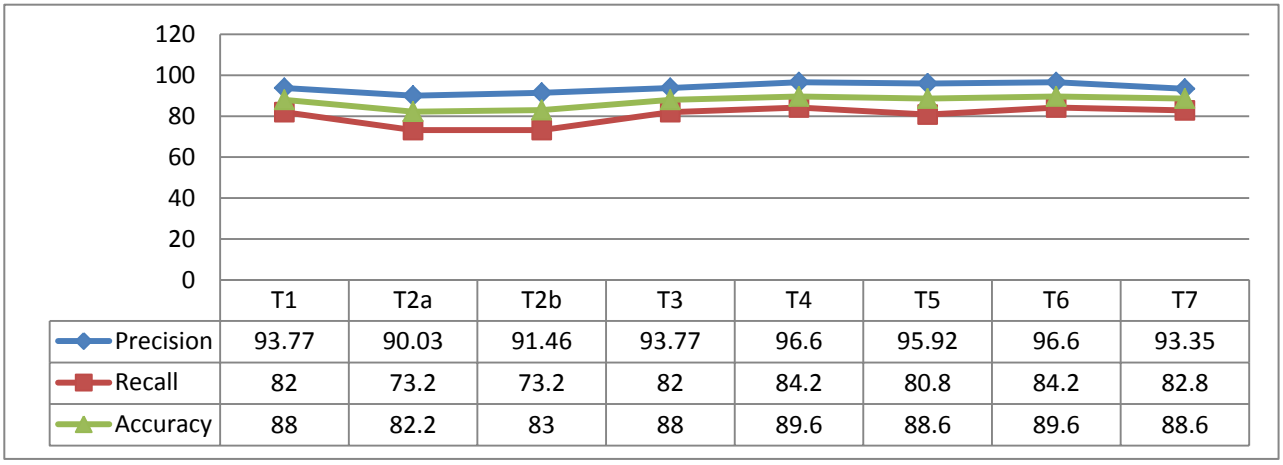
$$Precision = {}^{tp}/_{(tp + fp)} \quad (3)$$

$$Recall = {}^{tp}/_{(tp + fn)} \quad (4)$$

6.2. Performance of KNN Classifier on Movie Dataset

Figure 2 depicts the precision, recall and accuracy values for the KNN classifier for the 7 different representations. As it can be seen from the figure, KNN gives its best values for the three measures in Tier 4 and Tier 6. The common factor for these two tiers is that both use word n-grams (n=4). Tier 6 adds feature correlation for word n-grams. Feature correlation does not add value to the vectors when combined with word n-grams. The best value for recall was 84.2, for precision 96.6 and for accuracy 89.6. In other words, for KNN word n-grams and feature correlations enhance polarity detection. Oddly enough, stemming did not improve the results, as it can be seen in Tier 2a and Tier 2b. Tier 2a uses

stemming and Tier 2b uses light stemming. This could be contributed to the fact that built-in stemming algorithms of Rapidminer were used and these do not give high accuracy.

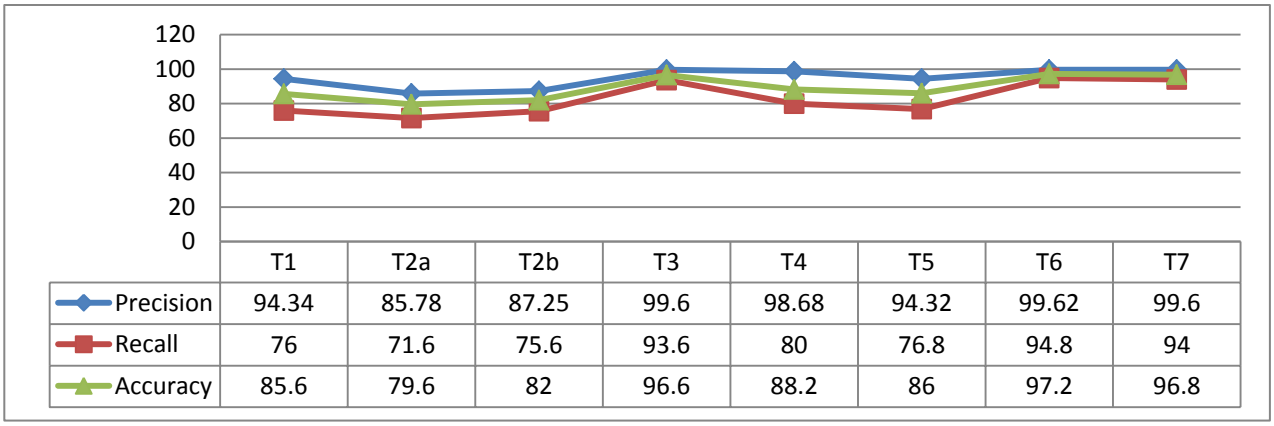


	T1	T2a	T2b	T3	T4	T5	T6	T7
Precision	93.77	90.03	91.46	93.77	96.6	95.92	96.6	93.35
Recall	82	73.2	73.2	82	84.2	80.8	84.2	82.8
Accuracy	88	82.2	83	88	89.6	88.6	89.6	88.6

Figure 2: Performance of KNN Classifier for the Movie Dataset.

6.3. Performance of Naïve Bayes Classifier on Movie Dataset

Figure 3 shows the performance of Naïve Bayes. It should be noted that among the three classifiers, Naïve Bayes benefited greatly from feature reduction. This is understandable as Naïve Bayes assumes independence between features and feature reduction retains only features that have low correlations with each other. The Naïve Bayes achieves its highest performance when word n-grams are combined with feature reduction (i.e. Tier 6). In Tier 6, precision equals 99.62, recall equals 94.8, and accuracy 97.2. These results are exceptionally good results.

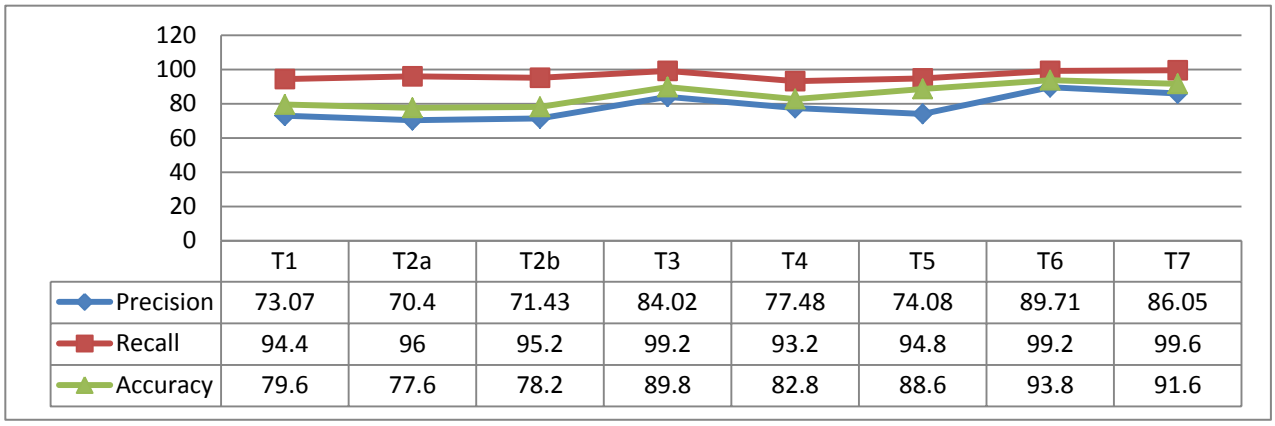


	T1	T2a	T2b	T3	T4	T5	T6	T7
Precision	94.34	85.78	87.25	99.6	98.68	94.32	99.62	99.6
Recall	76	71.6	75.6	93.6	80	76.8	94.8	94
Accuracy	85.6	79.6	82	96.6	88.2	86	97.2	96.8

Figure 3: Performance of Naïve Bayes for Movie Dataset.

6.4. Performance of SVM on Movie Dataset

Figure 4 shows the performance of SVM with the different representation strategies. The behavior of the SVM classifier is similar to the behavior of the other two classifiers. Precision improves with word n-grams combined with feature reduction. Recall improves with both character and word n-grams combined with feature reduction. Also, the accuracy reaches its maximum (93.8%) with word n-grams combined with feature reduction (T6) .



	T1	T2a	T2b	T3	T4	T5	T6	T7
Precision	73.07	70.4	71.43	84.02	77.48	74.08	89.71	86.05
Recall	94.4	96	95.2	99.2	93.2	94.8	99.2	99.6
Accuracy	79.6	77.6	78.2	89.8	82.8	88.6	93.8	91.6

Figure 4: Performance of the SVM Classifier for the Movie Dataset.

6.5. Performance for the Politics Dataset

The same set of experiments was carried out on the Politics dataset. The best set of features was determined experimentally for every classifier. In the case of Naïve Bayes and unigrams, the number of features was 76. For the SVM and unigrams, the number of features was 301. Finally, for K-NN, the number of optimal features for unigrams was also 76. Furthermore, the optimal number of n-grams in the case of word n-grams and character n-grams were determined experimentally. For example, the optimal value of n for Naïve Bayes and SVM was 3 words. Table 6 shows the accuracy values for the three classifiers at every tier with the optimal number of features and n-grams.

As it can be seen from Table 6, Tier 1 or baseline vectors, did not give good accuracy. In fact, the highest accuracy was 59.82% obtained by using SVM on the Politics dataset. Tier 2a (Stopword removal + Stemming) enhances the accuracy of all classifiers. For example, the accuracy of Naïve Bayes equals 58.87 compared to 53.80%. SVM accuracy also increased to reach 62.99%.

Tier 2b (Stopword removal + Light stemming) actually reduces the accuracy compared to tier 2a. Therefore, for the Politics dataset, stemming was more helpful than light stemming.

Tier 3 (Word vectors + Feature correlation) immensely enhanced the Naïve Bayes accuracy to reach 78.64% with optimal feature number equal to 76 features. The accuracy of SVM equal 73.90% at 301 features. By comparison, the accuracy of KNN is slightly reduced to reach 62.11% at 76 features.

Tier 4 (Word n-grams) and Tier 5 (Character n-grams) did not give better results than Tier 3 for the Politics dataset and SVM and Naïve Bayes. However, Tier 4 and Tier 5 enhanced the accuracy of KNN.

Tier 6 and Tier 7, by comparison, enhanced the accuracy of Naïve Bayes and SVM but did not enhance the accuracy of KNN.

To summarize the finding listed in Table 6, Naïve Bayes reaches its highest accuracy at Tier 7 with accuracy equal 85.70%, SVM reaches its highest accuracy at Tier 7 also at accuracy equal 82.47%, and KNN reaches its accuracy at Tier 4 with accuracy equal 66.67%

6.6. Remarks on Obtained Results

By revisiting our results we can conclude that preprocessing, as described in this research, enhances the classifiers' accuracy. All preprocessing techniques overcome the base case (when no preprocessing was used). There was one exception, stemming and light stemming for the Movie dataset – where the base case representation outperforms the accuracy of classifiers when stemming and light stemming were used. The reason for this is that the errors produced by the stemming algorithm for the Movie dataset produced too many features and increased the dimensionality of the dataset and this has affected the accuracy of the three classifiers. On the other hand, the accuracy has increased when stemming was carried out to the Politics dataset. This behavior could be explained as this dataset has fewer numbers of

features with great overlap between the reviews and therefore stemming, in this case, was useful despite the poor stemming results.

Table 6: Accuracy for the Politics Dataset

Data Representation	Naive Bayes	SVM	K-NN
Tier 1: Word Vectors	53.80%	59.82%	59.17%
Tier 2a: Word Vectors + Stop Words Removal + Root Stemming	58.87%	62.99%	62.36%
Tier 2b: Word Vectors + Stop Words Removal + Light Stemming	58.58%	61.13%	60.12%
Tier 3: Word Vectors Representation + Feature Correlation Reduction	***Features = 76***	***Features = 301***	***Features = 76***
	78.64%	73.90%	62.11%
Tier 4: Word Vectors Representation + Word n-Grams	***N=3***	***N=3***	***N=5***
	56.34%	59.85%	66.67%
Tier 5: Word Vectors Representation + Character n-Grams	***N=3***	***N=3***	***N=3***
	58.61%	61.72%	63.99%
Tier 6: Word Vectors + Root Stemming + Word n-Grams + Feature Correlation Reduction	***N=3***	***N=3***	***N=4***
	Features = 301	***Features = 301***	***Features = 76***
	80.88%	79.90%	62.42%
Tier 7: Word Vectors + Root Stemming + Character n-Grams + Feature Correlation Reduction	***N=3***	***N=4***	***N=4***
	Features = 301	***Features = 376***	***Features = 76***
	85.70%	82.47%	64.67%

Moreover, the substantial increase of the accuracy has been seen mainly when the features are reduced according to their correlation to the two class labels. Naïve Bayes was the most influenced classifier by this technique by achieving 96.62% accuracy with the Movie dataset and 78.64% with the Politics dataset compared to the baseline accuracies which were 85.60% and 53.80% respectively. The reason behind these performance enhancements is because the feature correlation reduction technique improves the independence between the features. The SVM classifier has also got a significant increase in the accuracy as this feature reduction technique helped in pruning non-relevant features and most probably could have reduced number of noise examples that would affect the process of identifying an optimal hyper plane and consequently affect the performance of SVM. The K-NN classifier, however, did not show remarkable benefit from feature reduction. This classifier's accuracy did not change in the case of the Movie dataset and only slight increment is obtained when applied on the Politics dataset. This is because the K-NN is influenced only with the majority of the closest K reviews to the one being tested. Adding a new review to the training examples could change the overall behavior of this classifier.

Finally, the use of n-grams in its both levels, words and characters, improved the performance of the three classifiers. The exception is KNN for the Politics dataset. This can be seen clearly in the use of word grams as it impose order of words and helps more in identifying negations when expressing opinions. The character n-grams, on the other hand, enforces uniform representation of the whole review document by arranging it into sequence of characters which are now treated as the new features and this helps in improving the accuracies.

7. Conclusions and Future Work

This paper has addressed sentiment analysis for reviews expressed in the Arabic language. Two datasets were used for the experiments carried out in this research. 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 Aljazeera website. The reviews of the Politics dataset tend to be short, informal and written by the general public. The other dataset, by comparison, is called the Movie dataset [23, 24] and is publically available. It consists of 500 reviews: 250 positive reviews and 250 negative reviews. These reviews were created by expert reviewers and uses modern standard Arabic.

Several aspects of data representations were investigated in this work. The feature vectors of the reviews were preprocessed in several ways and the effects of these on the classifiers' accuracy were investigated. The results show that stemming and light stemming combined with stopwords removal adversely affected the performance of the classification for the Movie dataset and slightly improves the classification for the Politics dataset. This is related to the fact that the stemming algorithms used were the ones that come free with Rapidminer and these have high error rates. In

the case of the Politics dataset, the number of features was small and therefore the errors in stemming have limited effects on the accuracy.

Results also show that feature reduction, by keeping only features that are highly correlated with the classes and less correlated with each other, improved the classification accuracy for the three classifiers and the two datasets. The accuracy of the Naïve Bayes classifier greatly improved by feature reduction as this enhances independence among the features.

Finally, word and character n-grams improved the results as well. N-grams helped in capturing the negated phrases and common phrases that are used in expressing sentiment. The results were biased for word n-grams over character n-grams for the Movie dataset.

Overall, the figures obtained for the Movie dataset were far better than the figures of the Politics dataset as the Movie dataset includes formal reviews that were written by educated and expert reviewers. The reviews in the Politics dataset were short and informal.

The performance of the classifiers was dependent on the preprocessing strategy and the dataset. For example, Naïve Bayes accuracy reached 96.6% for the Movie dataset when correlated features were used as this enhances the independence between features of the reviews which aligns with the independence assumption of Naïve Bayes. Character n-grams as a preprocessing strategy worked well for both SVM and KNN, both have accuracies equal 88.60% for the Movie dataset. Word n-grams boosted the accuracy of K-NN where it reached 89.6% for the Movie dataset. For the Politics dataset, Naïve Bayes gave the highest accuracy (85.7%) when compared to the other two classifiers (SVM 82.47% and KNN 66.67%).

This work has opened many venues for future research. For example, the work reported here deals with sentiment analysis at the review or document level. A useful extension would be to find the sentiment at the sentence level. Furthermore, the work could be extended to deal with finding opinions about aspects (of the products).

Notes

1. http://rapid-i.com
2. http://www.Aljazeera.net

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

[1] Han J and Kamber M. Data Mining Concepts and Techniques. 2nd Edition. San Francisco, CA: Morgan Kaufmann Publishers, 2006.
[2] Tan P N, Steinbach M and Kumar V. Introduction to Data Mining. Boston, MA: Addison Wesley, 2005.
[3] Aue A and Gamon M. Customizing Sentiment Classifiers to New Domains: A Case Study. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP*. Borovets, BG, 2005 .
[4] Go A, Bhayani R and Huang L. Twitter Sentiment Classification Using Distant Supervision. Processing 2009; 150(12): 1-6.
[5] Pang B and Lee L. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval 2008; 2(1-2): 1-135.
[6] Read J. Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification. In: *Proceedings of the ACL Student Research Workshop*. Stroudsburg, USA, 2005, pp. 43-48.
[7] Khan, K, Baharudin B, Khan A, Malik F. Mining Opinion from Text Documents: A Survey, In: *Proceedings of the 3rd IEEE International Conference on Digital Ecosystems and Technologies*. Istanbul, 2009, pp. 217 – 222.
[8] Web. Top 30 Languages of the World. http://www.vistawide.com/languages/top_30_languages.htm. Last Accessed: 3 Sep. 2013.
[9] Saleh MR, Martín-Valdivia MT, Ureña-López LA and Perea-Ortega JM. OCA Corpus English Version. http://sinai.ujaen.es/wiki/index.php/OCA_Corpus_(English_version). Last Accessed 3 Sep. 2013.
[10] Wikipedia. Sentiment Analysis. http://en.wikipedia.org/wiki/Sentiment_analysis. Last Accessed 3 Sep. 2013.
[11] Pang B, Lee L and Vaithyanathan S. Thumbs up? Sentiment Classification using Machine Learning Techniques. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '02).* Morristown, 2002, pp. 79–86.
[12] Bickerstaffe A and Zukerman I. A Hierarchical Classifier Applied to Multi-way Sentiment Detection. In: *Proceedings of the 23rd International Conference on Computational Linguistics.* Beijing, China, 2005, pp. 62-70.
[13] Paltoglou G and Thelwall M. A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Uppsala, Sweden, 2010, pp. 1386–1395.

[14] Hall M. Correlation-based Feature Selection for Machine Learning. *PhD Thesis*, University of Waikato, 1999.
[15] Esuli A and Sebastiani F. Determining Term Subjectivity and Term Orientation for Opinion Mining. In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, IT, 2006, pp. 193– 200.
[16] Sarvabhotla K, Pingali P and Varma V. Sentiment Classification: A Lexical Similarity Based Approach for Extracting Subjectivity in Documents. Information Retrieval 2011; 14(3): 337– 353.
[17] Denecke, K. Are SentiWordNet Scores Suited for Multi-domain Sentiment Classification? In: *Proceedings of the Fourth International Conference on Digital Information Management (ICDIM).* Ann Arbor, MI, 2009, pp. 33– 38.
[18] Ohana B and Tierney B. Sentiment Classification of Reviews using SentiWordNet. In: *Proceedings of the 9th. Information Technology and Telecommunication Conference (IT & T).* Dublin, Ireland, 2009.
[19] Esuli A and Sebastiani, F. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC).* Genova, Italy, 2006, pp. 417– 422.
[20] Denecke K. Using SentiWordNet for Multilingual Sentiment Analysis. In: *Proceedings of IEEE 24th International Conference on Data Engineering Workshop (ICDEW).* Hannover, Germany, 2007, pp. Pages 507-512.
[21] Kosinov, S. Evaluation of N-Grams Conflation Approach in Text-Based Information Retrieval. In: *Proceedings of International Workshop on Information Retrieval*. Edmonton, Alberta, Canada, 2001, pp. 136– 142.
[22] El-Halees A. Arabic Opinion Mining Using Combined Classification Approach. In: *Proceedings of the International Arab Conference on Information Technology (ACIT)*. Riyadh, Saudi Arabia, 2011.
[23] Thelwall M, Buckley K, Paltoglou G, Cai D, and Kappas A. Sentiment Strength Detection in Short Informal Text. Journal of the American Society for Information Science and Technology 2010; 61(12): 2544– 2558.
[24] Ahmad K and Almas Y. Visualizing Sentiments in Financial Texts. In: *Proceedings of the Ninth International Conference on Information Visualization.* Washington, USA, 2005, pp. 363 – 368.
[25] Ahmad K, Cheng D, and Almas, Y. Multi-lingual Sentiment Analysis of Financial News Streams. In: *Proceedings of the 1st International Workshop on Grid Technology for Financial Modeling and Simulation*. Palermo, Italy, 2006.
[26] Abbasi A, Chen H and Salem A. Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. ACM Transactions on Information Systems (TOIS) 2008; 26(3): 1 – 34.
[27] Elhawary M and Elfeky M. Mining Arabic Business Reviews. In: *Proceedings of the IEEE International Conference on Data Mining, Dec, Mountain View.* USA, 2010, pp. 1108 – 1113.
[28] Saleh MR, Martín-Valdivia MT, Ureña-López LA and Perea-Ortega J M OCA: Opinion Corpus for Arabic. Journal of the American Society for Information Science and Technology 2011; 62(10): 2045– 2054.
[29] Rahmoun A and Elberrichi Z. Experimenting N-Grams in Text Categorization. The International Arab Journal of Information Technology, 2007; 4(4): 377 – 385.
[30] Wu X. et al. Top 10 Algorithms in Data Mining, Knowledge and Information Systems 2008; 14(1): 1– 37.