EXTENDING THE KNOWLEDGE OF THE ARABIC SENTIMENT CLASSIFICATION USING A FOREIGN EXTERNAL LEXICAL SOURCE

Saud S. Alotaibi[1] and Charles W. Anderson[2]

[1]Information Technology Deanship,Umm AlquraUniverisity,Makkah, Saudi Arabia
[2]Computer Science Department,Colorado State University,Fort Collins, USA

ABSTRACT

This article introduces a methodology for analyzing sentiment in Arabic text using a global foreign lexical source. Our method leverages the available resource in another language such as the SentiWordNet in English to the limited language resource that is Arabic. The knowledge that is taken from the external resource will be injected into the feature model whilethe machine-learning-based classifier is trained. The first step of our method is to build the bag-of-words (BOW) model of the Arabic text. The second step calculates the score of polarity using translation machine technique and English SentiWordNet. The scores for each text will be added to the model in three pairs for objective, positive, and negative. The last step of our method involves training the ML classifier on that model to predict the sentiment of the Arabic text. Our method increases the performance compared with the baseline model that is BOW in most cases. In addition, it seems a viable approach to sentiment analysis in Arabic text where there is limitation of the available resource.

KEYWORDS

Sentiment analysis, natural language processing,Arabic sentiment classification,machine translation,machine learning

1. INTRODUCTION

The extraction of sentiment from a text has attracted a considerable amount of attention over the past decade, both in the industry and academia. Sentiment analysis attempts to extract the emotions and opinions of individuals from their writing about specific entities. Much of the research has been undertaken in English as this is the dominant language of science. Arabic natural language processing has therefore become attractive to researchers because of its complexity and the scarcity of available resources. According to Farghaly and Shaalan in [1], the field of natural language processing (NLP) in Arabic is still at an early stage of evolution despite the efforts being made with the fundamental NLP tools of Arabic.

Sentiment analysis (SA) of Arabic is also still in its early stages [2], and increased effort and reliability of low-level tools are required in order to build upon this foundation. Relying only on the actual word to build a feature model is a good starting point in sentiment analysis, but there is a need to add more information about the text in the feature model. The semantic orientation technique[3] in sentiment analysis only relies on calculating the polarity score of each word in the document. After that, the final decision about the text's sentiment is taken depending on the calculated value. When the polarity score is added to the feature model, this approach may get the benefit of semantic technique and merge it with the ML method. Therefore, adding this feature leads us to the hybrid method when the ML technique is used as a primary classifier and supports it with some of the semantic orientation concept.

DOI: 10.5121/ijnlc.2016.5301

The rest of this article is organized as follows: The second section shows the related work. The third section explains our polarity orientation method for Arabic sentiment analysis. The fourth section describes the experiment that is carried out and discusses the results. The last section concludes the research and illustrates some future work.

2. RELATED WORKS IN ARABIC SENTIMENT ANALYSIS

Much of sentiment analysis research has been done in English as this is the dominant language of science. Recently, a few researchers have concentrated on applying sentiment analysis to other languages, one such language being Arabic. This section shows that related works have been done in Arabic in different aspects including corpora, features, and methodologies.

A.Arabic Sentiment Corpora

The opinion corpus for Arabic (OCA) [4] (which is the only published corpus) contains 500 movie reviews. They are annotated at the document level. Half the reviews are considered positive, and the rest are negative. Further work,called AWATIF,has been undertaken to build a multi-genre subjectivity and sentiment corpus for modern standard Arabic [5]. The domain of this data was taken from a newswire in different domains (400 documents), Wikipedia talk pages (around 5,342 sentences), and web forums (around 2532 threads from seven web forums). The annotation was at the sentence level, and three different conditions were used to annotate the data: (1) Gold Human with Simple Guidelines (GH-SIMP), (2) Gold Human Linguistically-motivated and Genre-nuanced (GHLG), and (3) Amazon Mechanical Turk with Simple Guidelines (AMT-SIMP) [5]. In addition, the authors attempted to build a labeled social media corpus for subjectivity and sentiment in the Arabic language in the SAMAR project [6]. The data was collected from four different types of social media. These included Arabic chatting, tweets, Wikipedia talk pages, and forums. This corpus was a mix of long and short sentences, as well as MSA and some of DA. They provided standoff annotations on top of the Arabic Treebank (ATB)[1] part 1, version 3, which is only free for the user who has subscribed with the LDC[2] since 2003.

B.Features and Methods

Abbasi et al. [2] proposed a system for sentiment analysis task in a multi language web forum at document level. The system depends on an Entropy-Weighted Genetic Algorithm (EWGA) to choose the best features and the SVM with linear kernel for the sentiment classification. Their method tries to find an overlap between language-independent features, including syntactic and stylistic features. The syntactic features include POS only for the English language, not for Arabic. In order to evaluate the performance of their method, the authors measured the accuracy of the classifier by dividing the number of correctly classified documents by the total number of documents. In this case, a more accurate measurement was required to help evaluate the method in both classes. The authors reported that syntactic features achieved a higher result than the stylistic ones. When the two features were employed together using EWGA, the accuracy result increased to 93.6% in the Middle Eastern forum domain.

The work of Rushdi-Saleh et al. [4] focused on investigating two ML classifiers, Naive Bayes and Support Vector Machine, with two different weighting schemes (term frequency and term frequency–inverse document frequency) and three n-gram models. The effect of using the stem of the Arabic work was also investigated with different n-gram models. The authors built their sentiment corpus by collecting around 500 Arabic movie reviews from different websites. They reported an accuracy of 90.6% using the SVM with the tri-gram model and with no stemming for document-level classification. In addition, they claimed that there was no big impact of using TF

[1]http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2005T02
[2]http://www.ldc.upenn.edu/

or TF-ID as a weighting scheme, which makes sense because both schemes represent the count of the term over the document. It could be useful to compare the presence of the term versus the term-frequency scheme.

El-Halees [7] proposed a combined classification approach for document-level polarity classification in Arabic. His method applied three different classifiers in a sequential manner: a lexicon-based classifier, a maximum entropy classifier, and the K-Nearest Neighbor classifier. The result from one classifier was used as training data for the next. The text was manipulated before using the first classifier by removing the stop words. Some Arabic letters were normalized, and some misspelled words were corrected. A simple stemmer was used here to generate the stem of the Arabic words, and TF-IDF was used as the term-weighting scheme. The F-measure was used as the evaluation metric. The F-measure that was reported in this method was between 75% and 84% depending on the domain of the data. The average of the F-measure was also calculated, 82% for the positive document and 78% for the negative one. The main issue for this study was that there were no more features added to the classifier that could help increase the performance and accuracy.

Other studies have attempted to investigate the linguistic features of Arabic and combine these with an ML classifier in order to perform sentiment analysis. One such study tried to analyze the grammatical structure of Arabic [8]. It attempted to analyze the sentiment at the sentence level first and then use the results to analyze the sentiment at the document level. At the sentence level, the researchers compared two different approaches. The first was generalizing the Arabic sentence into a general structure that contains the actor and the action. The second approach used some semantic and stylistic features. The researchers used different classifiers for a different approach. They used the SVM for the grammatical classifier and obtained an accuracy of 89%, while the J48 decision tree was used with the semantic approaches and achieved an accuracy of 80% when the semantic orientation of the words extracted and assigned manually were used and 62% when the dictionary was used.

Another work, which investigated the effect of language-independent and Arabic-specific features on the performance of the classifier, was conducted by Abdul-Mageed et al. [6]. They performed two kinds of sentence-level sentiment analysis for two different domains: news and social media. The SVM was used to classify both the subjectivity and polarity of the sentences with different features, including N-gram, adjective features and a unique feature where all words occurring fewer than four times were replaced by the token “UNIQUE,” as well as MSA morphological features (person, gender, and number). Using different stemming and lemmatization settings with different types of independent language and Modern Standard Arabic morphology features, the researchers achieved an F1 result of 72% for subjectivity and 96% for polarity with stem, morphology setting, and ADJ features using the newswire domain. In SAMAR [6], they investigated the effect that the standard features and the genre-specific features had on the subjectivity and sentiment classification of the Arabic social media domain.

C.Polarity Score as a Feature

In the Arabic language, there is a lack of these resources in the case of adding the polarity score to the sentiment analysis. Some of them are not available for free or are incomplete. Abdul-Mageed et al. [9] manually built an Arabic lexicon comprising a list of approximately 4,000 Arabic adjectives from the newswire domain annotated for polarity. This corpus only contains one type of POS, adjectives, and is not comparable with the English SentiWordNet. It is only a collection of the positive and negative words without any of the scoring values. Recently, Alhazmi et al. [10] discussed the issue of building the Arabic SentiWordNet and started to put the first step in place to create this corpus. However, they are still working on it in order to enhance its performance before making it free publicly.

3.PROPOSED METHOD

Relying only on the actual word to build a feature model is a good starting point in sentiment analysis, but there is a need to add more information about the text in the feature model. Our proposed method may relate to the semantic orientation approach of the sentiment analysis techniques.The semantic orientation technique in sentiment analysis only relies on calculating the polarity score of each word in the document. After that, the final decision about the text's sentiment is taken depending on the calculated value. When the polarity score is added to the feature model, this wouldget the benefit of semantic technique and merge it with the machine learning "ML" method. Therefore, adding this feature results in a hybrid method when the ML technique is used as a primary classifier and supports it with some of the semantic orientation concept.

Our method is based on the calculation of the score of polarity of each text. In order to obtain the value of the word *polarity*, a lexical resource, such as SentiWordNet, is needed. This lexical resource is a corpus-based lexical resource constructed from the perspective of WordNet. Each synset, which is a set of one group of synonyms, is assigned three sentiment scores: positivity, negativity, and objectivity[12]. There is a leak of this kind of corpus for Arabic language. Until the time of preparing this work, there had been no available Arabic SentiWordNet. In French language, Ghorbel and Jacot in [13] found that using a machine translation to obtain the polarity score improved the performance of sentiment analysis.Therefore, we rely on the best state of the art technology that we have, which is the current English SentiWordNet[3] with the machine translation mechanism.

Figure 1 illustrates the details of how the polarity component works with Arabic sentiment analysis. The polarity score calculation part is responsible for calculating the score of a given word. In order to calculate the polarity score, we have to have a SentiWordNet corpus. In our case, this corpus is nonexistent. Therefore, we rely on an alternative approach. We believe that the optimal solution is the one that has the native Arabic SentiWordNet corpus. However, relying on a mature SentiWordNet in other language and using machine translation mechanisms might help to evaluate this approach.

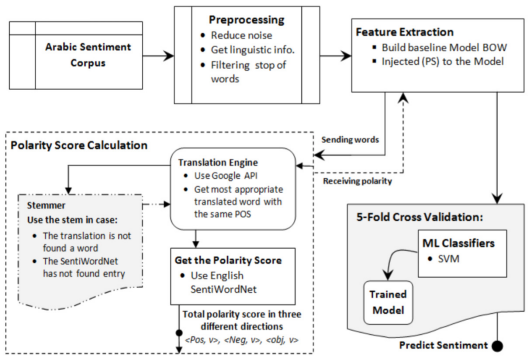

Figure 1.Details of the Polarity Score Calculation Component with Arabic Sentiment Analysis

In order to build and use this component with Arabic sentiment analysis processing, we download the latest version of the English SentiWordNet corpus. The other part of this component is the translation unit. We rely on this unit as one of the most mature translation servicesprovided by

[3] http://sentiwordnet.isti.cnr.it/

Google Translate API. The polarity soccer is then calculated for each record while building the feature model. In the case of document classification, for each document, the total polarity score should be calculated. This leads to three feature columns in the space model: one for positive, negative, and objective. For example, we have document d that has 40 words. Our approach will be to go over it, translate each of the document's words, and get the score from the English SentiWordNet. The total of the three feature vectors will be calculated at the end. In the end, we should have the following feature for the document d:

{d: [positive, value], [negative, value], [objective, value]}.

We rely on two different mechanisms in this proposed feature. The first one depends on computing the score of the polarity of the text. As explained earlier, the polarity would be calculated in the particular text for all words in that text. This score would be categorized into four different types: positive, negative, neutral, and objective polarity. The second mechanism is the one that relies on just counting the number of polar words. For each type of polarity, we count the words in each sentence. For particular texts, the polarity of each word should be known through the SentiWordNet, and we count the number of positive, negative, and objective words that we found in each text. Algorithm 1 explains the pseudo code of our approach.

Some research studies use a similar idea to our approach to Arabic language [5, 6, 9]. However, our approach is different from others in various aspects. First, we use the polarity score instead of using the actual polarity words and build the feature depending on that.This will be explained in detail in the next section. Abdul-Mageed in [11] started to build Arabic lexical corpus from different domains. Hislexical only contains the actual words and classifies them into different polarity categories. That means hiswork has only the words without any scoring value [11]. After that, he uses this lexical as a dictionary to build a feature model instead of using all words in the actual dataset of the main sentiment corpus. In our approach, we rely on the value of the polarity instead of the polarity type of the word only besides using the traditional bag-of-words model.

1. For each words *W* in the text *T*, do the following:
   - 1.1. Translate the actual word using translation engine: *W* would be *TW*,
   - 1.2. If *W* is not translated, then translate the stem of word *W* to set TW.
   - 1.3. Get the polarity score of *TW* from English SentiWordNet. The output would be {Positive: S, Negative: S, and Objective: S}.
   - 1.4. Determine the polarity type of *TW* and its score:
     - 1.4.1. If the positive score > negative and positive > = objective, then *TW* is positive with that score.
     - 1.4.2. Else if the negative score > positive and negative > = objective, then *TW* is negative with that score.
     - 1.4.3. Else if the objective score > positive and objective > negative, then *TW* is objective with that score.
     - 1.4.4. Else*TW* is neutral with that score.
2. **Compute the Polarity (Counting or Scoring)** as follows:
   - 2.1. Let assume that:
     - 2.1.1. Text *T* has words vector *W* $\{w_1, \dots, w_n\}$.
     - 2.1.2. We have four different polarity types *PT* {positive, negative, neutral, and objective}.
   - 2.2. For every polarity in *PT*, the *PolarityCount* or *PolarityScore* will be calculated as follows:
     - 2.2.1. $PolarityCount = \sum_{i=1}^{n} PolarityType(w_i)$,
       where *w* represents all words in the text, and *PolarityType* is the function that returns the polarity types of the word that is either positive, negative, or objective.
     - 2.2.2. $PolarityScore = \sum_{i=1}^{n} Polarity(w_i)$
       where *w* represents all words in the text, and *Polarity* is the function that calculates the polarity.

Algorithm 1. The steps of calculating polarity

In some cases, the translation is not able to translate some words. This issue makes us use other techniques with this approach to help and reduce the number of untranslated words. The stem mechanism is used to minimize the variation of the word and might preserve the semantic meaning of the word [1]. Therefore, we follow three techniques to investigate the effect of using stem with our approach. The first method uses only the actual word without applying the stem that is called "PolNoStem." The second technique is "PolWithStem." The word is first translated. When there is no translation found, it then transfers to apply the stem to it. After that, the stem word is translated."PolStemOnly" is the third technique. The stem of the word is extracted first, and then the stem will be translated and used to get the score of the polarity.

In the literature, there are three different root libraries for the Arabic language: Khoja Arabic stemmer [14],ISRI stemmer [15] and Tashaphyne Light Arabic stemmer [16]. The most suitable root library is Tashaphyne [16] because it has a real implementation using Python and can be used with the other tools that are utilized in the classification process.

4. EXPERIMANTAL RESULTS

This section has three parts. The first part describes the data that have been used. The second part illustrates the process that has been performed to test out proposed methods. The results are discussed in the last section.

A.Arabic Sentiment Corpus

The authors of this work build their own corpus because of the scarcity of sentiment Arabic corpus. The research corpus is built from five different genres, which include newswire, reviews on that news, market reviews, restaurants reviews, and movie reviews. The newswire data has been taken from the Sabq[4] website among different domains, which are local, sport, economics, technology, and social news. The restaurant reviews have been taken from the work of [17], which captures the review of the user concerning restaurants.[5] The movie reviews have been taken from the movie review website[6] and is used in [8]. The Souq[7] (considered as the Amazon marketplace for Arab countries) is used as a source for market reviews.In total, our corpus contains 6,268 documents with more than 33,000 sentences. There are around 7,674 positive sentences, 9,202 negative sentences, and 3,351 neutral sentences.

Two Arabic educated individuals have been chosen to annotate the data. Each annotator was given guidelines. First, they should determine if the document is subjective or objective. Second, they need to establish the polarity of the subjective text among three categories, these being positive, neutral, or negative. Third, the annotator should go over each sentence in the document, noting its polarity if the sentence is a subjective one; otherwise, the sentence should be seen as objective. The first step was to train the two annotators, who were then asked to work on the same dataset, which contained around 33% of the sentences. During this process, the inter-annotator agreement between them was calculated using the Kappa coefficient [18], which was between 0.72 and 0.84. In order to get these datasets, contact the first author.

[4] http://sabq.org
[5] http://www.qaym.com
[6] http://www.filfan.com
[7] http://saudi.souq.com/sa-ar

B.Classification Process

The preprocessing phase contains some steps before the text is passed to the classifier. The first step includes the filtering out of all rubbish data that might be found in the text, including single letters or non-Arabic characters. The second step is to normalize long words that may make some letters redundant. The third step is to use the AMIRA [19] toolkit for all data in order to prepare the part of speech tag of the words. The final step involves removing the stop word lists and modifying so as to deal with these while they build the vector space model that represents the words. Stop word lists in [20] were used. To evaluate our method, many experiments were undertaken using a support vector machine (SVM) classifier with linear kernel with fivefold cross-validation using scikit-learn library [21]. In our case, we divide our dataset into five disjoint parts with equal proportions of samples in each class. Four of them are used to train the classifier, while the rest will be used to test the model that is generated during the training process. This process will be repeated five times because we have five partitions of the data. Every time, a new partition is used for the testing phase. During every cycle, the F1 metric is calculated, which measures the accuracy of the classifierafter computing the precision and recall. We use the default parametersfor SVM that comes with the scikit-learn tool since we found that these parameters work well with our data. We also tried using linear and nonlinear kernel (RBFs) for the SVM, and we figured out that the linear kernel outperforms the nonlinear one in most cases, so we use the linear on our experiments.

As a baseline model, the unigram model is applied. The bi-gram or the tri-gram was not used since we only need to investigate the effect of adding polarity feature to the baseline model. If we add bi-/tri-gram models, we also need to calculate the actual polarity score for the gram unit. For example, if we add the bi-gram model to the feature, we need to recalculate the polarity for each bi-gram pair in the feature model. This might be investigated as a future work. The experiment is performed on two different levels: the sentencelevel and the document level.

C.Results and Discussion

Table 1 shows the results of using polarity approach at the sentence-level classification. The document-level classification results of this approach are displayed in Table 2. Different configurations and combinations are used while injecting the polarity concept with the feature model. The first row in these tables for each dataset represents the results of the baseline model feature that is BOW. The next three rows show a different configuration of using polarity concept with the feature model. PolNoStem refers to the method when the polarity is computed without using stem technique as it is explained early in section three. The “PolWithStem” represents using the stem when the word is not found in either translation corpus or SentiWordNet corpus. The row shows the method of polarity that is applied using the stem at the first stem. For each type of classification, either subjectivity or polarity classification, two mechanisms are used to add polarity into the feature model. The first column refers to the count method and the second for calculating the scoring of polarity as illustrated previously. The best results are written in boldface.

Table 1. Results Of Adding Polarity Score And Count As A Feature In Arabic Sentence-Level Classification

		Subjectivity		*Polarity*	
		Count	***Score***	***Count***	***Score***
News Reviews	BOW	69.2%	69.2%	58.1%	58.1%
	PolNoStem	70.1%	70.1%	**58.3%**	58.0%
	PolWithStem	**70.4%**	70.1%	58.0%	**58.3%**
	PolStemOnly	70.2%	**70.5%**	58.2%	58.2%
Restaurant Reviews	BOW	71.0%	71.0%	83.4%	**83.4%**
	PolNoStem	71.4%	71.3%	83.5%	83.3%
	PolWithStem	**71.5%**	**71.4%**	**83.9%**	83.3%
	PolStemOnly	71.3%	**71.4%**	83.3%	**83.4%**
Market Reviews	BOW	**89.3%**	**89.3%**	88.2%	88.2%
	PolNoStem	89.0%	**89.3%**	**88.3%**	88.3%
	PolWithStem	**89.3%**	**89.3%**	88.1%	88.1%
	PolStemOnly	89.1%	88.7%	88.0%	**88.4%**
Movie Reviews	BOW	**45.0%**	**45.0%**	80.0%	80.0%
	PolNoStem	44.7%	44.9%	**80.4%**	**80.8%**
	PolWithStem	44.6%	44.8%	79.4%	80.1%
	PolStemOnly	44.9%	44.9%	80.3%	80.0%
Newswire	BOW	35.2%	35.2%	80.1%	80.1%
	PolNoStem	35.8%	35.6%	81.3%	81.0%
	PolWithStem	35.7%	**36.3%**	**81.5%**	80.7%
	PolStemOnly	**36.0%**	36.0%	80.4%	**81.3%**

Key: "PolNoStem" indicates the F1-score of using polarity feature without using the stem method with the baseline model, "PolWithStem" shows the results of using polarity feature with the stem method when the actual word does not have translation, and "PolStemOnly" displays the results using polarity feature and the stem of the word first before the translation.

Most of the time, using the polarity method outperforms the baseline model. The polarity approach helps more in the case of polarity classification type than the subjectivity one because it may add more detail about the polarity aspect than the subjectivity orientation. However, this method adds some performance and knowledge to the classifier in case of subjectivity. For example, the result increased by more than 1% in the case of subjectivity for the newswire domain with scoring technique. This trend of increasing the performance would be also found in all classification types (subjectivity or polarity) in Table 1. The same trend of improvement is also found in the document-level classification in Table 2. In general, the results improved by 0.5%.

Regarding the best stem configuration mechanisms that should be used with polarity method, we notice that using stem mechanism with polarity approach helps to improve the performance of the proposed model. This improvement is seen especially in the dataset domain that has dialect Arabic language, that is, the restaurant reviews. This might come from the nature of the dataset itself. We have different dialect words that might be derived from the MSA Arabic butare not found in MSA as an actual word. The translation engine only works well with the MSA Arabic. Therefore, using stem in some cases help the translation to find appropriate work in both types of Arabic language "MSA and DA." For example, the result increases by 0.5% with polarity classification in the restaurant review domain (Table 1).

Table 2.Results Of Adding Polarity Score And Count As A Feature In Arabic Document-Level Classification

		Subjectivity		*Polarity*	
		Count	***Score***	***Count***	***Score***
News Reviews	BOW	88.1%	88.1%	**56.4%**	56.4%
	PolNoStem	88.1%	88.1%	55.9%	56.4%
	PolWithStem	88.1%	88.1%	**56.4%**	55.8%
	PolStemOnly	**88.2%**	**88.2%**	55.9%	**56.5%**
Restaurant Reviews	BOW	**96.2%**	**96.2%**	**85.3%**	**85.3%**
	PolNoStem	95.9%	95.9%	84.5%	84.6%
	PolWithStem	95.9%	95.9%	84.4%	84.8%
	PolStemOnly	95.9%	95.9%	**85.3%**	84.7%
MarketReviews	BOW	**93.4%**	**93.4%**	90.0%	90.0%
	PolNoStem	**93.4%**	**93.4%**	90.3%	**90.5%**
	PolWithStem	93.1%	93.1%	**90.5%**	89.9%
	PolStemOnly	93.2%	93.2%	89.9%	**90.5%**
Movie Reviews	BOW	NA	NA	**80.0%**	80.00%
	PolNoStem	NA	NA	78.1%	79.2%
	PolWithStem	NA	NA	79.0%	**80.1%**
	PolStemOnly	NA	NA	78.1%	79.2%
Newswire	BOW	63.4%	63.4%	76.4%	**76.4%**
	PolNoStem	64.3%	63.6%	75.6%	**76.4%**
	PolWithStem	**64.8%**	62.5%	**76.9%**	75.6%
	PolStemOnly	63.4%	**64.2%**	74.5%	75.2%

Key: "PolNoStem" indicates the F1-score of using polarity feature without using the stem method with the baseline model, "PolWithStem" shows the results using polarity feature with the stem method when the actual word does not have translation, and "PolStemOnly" displays the results using polarity feature and the stem on the word first before the translation.

In order to make a judgment on the best polarity techniques (counting or scoring) that should be used, we calculate which method achieves the best result in each classification process. Table 3 illustrates this comparison. For example, the polarity counting method achieves the best result three times compared with nine times for the polarity scoring method using the "PolNoStem" feature model in all classification types in the document-classification level. We have noticed that the scoring technique outperforms the counting in the case of document-level classification with 20 times versus 13 times. On the other hand, the counting achieves 19 best results versus 14 in the case of the sentence-level classification. This suggests that counting polar words is better than calculating their score in the case of the sentence-level classification. That means the score value of the total polarity only works best for the long text, and the counting method works best for the short text such as the sentence. The other observation that we can infer from the data in Table 3 is the counting technique works better with the "PolWithStem" model than the Scoring. The Scoring works well with applying stem first in the "PolStemOnly" model. This might help if we want to reduce the effect of the actual word and use the stem technique.

Table 3. Counting Versus Scoring In Each Document And Sentence Classification

Model	Document Level Classification		Sentence Level Classification	
	Counting	Scoring	Counting	Scoring
PolNoStem	3	9	9	2
PolWithStem	9	2	9	4
PolStemOnly	2	9	1	9

Key: "PolNoStem" indicates the F1-score of using polarity feature without using the stem method with the baseline model, "PolWith Stem" shows the results using polarity feature with the stem method when the actual word does not have translation, and "PolStemOnly" displays the results using polarity feature and the stem on the word first before the translation. "Counting" indicates the method of counting the polar words, whereas "Scoring" shows the method of calculating the polarity score.

5. CONCLUSION AND FUTURE WORK

In this progress work, the sense of polarity semantic is added to the Arabic sentiment analysis. Our proposed method is adding global sentiment orientation knowledge to the classifier instead of relying on the BOW model that builds from the same domain. Counting and scoring are two different orientations of the polarity that are involved in our method. In addition, the stem technique is used to leverage the performance of our method and reduce some of the weakness of using foreign SentiWordNet corpus. The results show that the proposed method is promising. The performance was not significant compared with the baseline model. This issue comes from applying the translation step and using English lexical corpus. In the future, this method could be used with other features such as the part-of-speech feature. Furthermore, we could apply this method on phrase level instead of using it on the word level. The phrases of the sentence are prepared first, and then we calculate the polarity score of each phrase. In the end, we can usethe actual score of the word instead of using term frequency to build the feature model.

6. REFERENCES

[1] A. Farghaly and K. Shaalan. 2009. Arabic Natural Language Processing: Challenges and Solutions. 8(4): 14: 1–14:22, December. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] A. Abbasi, Hsinchun Chen, and A. Salem. 2008. Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. ACM Trans. Inf. Syst., 26(3):12:1–12:34, June.

[3] P. Chaovalit and L. Zhou. 2005. Movie review mining: A comparison between supervised and unsupervised classification approaches. In System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on, p. 112c, Jan.

[4] M. Rushdi-Saleh, M. Martin-Valdivia, L. Urena-Lopez, and J. Perea-Ortega. OCA: Opinion corpus for Arabic. Journal of the American Society for Information Science and Technology, 62(10): pp. 2045–2054, 2011.

[5] M. Abdul-Mageed and M. Diab. AWATIF: A multi-genre corpus for modern standard Arabic subjectivity and sentiment analysis. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), pp. 19–28, Istanbul, Turkey, May 2012.

[6] M. Abdul-Mageed, S. Kubler, and M. Diab. Samar. A system for subjectivity and sentiment analysis of Arabic social media. In Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, pp. 19–28. Association for Computational Linguistics, 2012.

[7] A. El-Halees. Arabic opinion mining using combined classification approach.In Proceeding The International Arab Conference On Information Technology, Azrqa, Jordan, 2011.

[8] N. Farra, E. Challita, R. A. Assi, and H. Hajj. Sentence-level and document-level sentiment mining for Arabic texts.In Data Mining Workshops (ICDMW), 2010 IEEE International Conference on, Dec. 2010, pp. 1114 –1119.

[9] M. Abdul-Mageed, M. T. Diab, and M. Korayem. Subjectivity and sentiment analysis of Modern Standard Arabic.In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - volume 2, ser. HLT '11. Strouds- burg, PA, USA: Association for Computational Linguistics, 2011, pp. 587–591

[10] S. Alhazmi, W.J. Black, and J. McNaught. 2013. Arabic SentiWordNet in relation to SentiWordNet 3.0. International Journal of Computational Linguistics, 4(1):1– 11.

[11] Muhammad Abdul-Mageed and Mona Diab. 2012. Toward Building a Large-Scale Arabic Sentiment Lexicon. In Proceedings of the 6th International Global Word-Net Conference, Matsue, Japan.

[12] S. Baccianella, A. Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In LREC, 10, pp. 2200–2204.

[13] H. Ghorbel and D. Jacot. Further experiments in sentiment analysis of french movie reviews. In E. Mugellini, P. Szczepaniak, M. Pettenati, and M. Sokhn, editors, Advances in Intel- ligent Web Mastering, volume 86 of Advances in Intelligent and Soft Computing, pp. 19–28. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-18028-6.

[14] S. Khoja and R. Garside. Stemming Arabic text. Lancaster, UK, Computing Department, Lancaster University, 1999.

[15] K. Taghva, R. Elkhoury, and J. Coombs. Arabic stemming without a root dictionary. In Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, volume 1, pp. 152–157. IEEE, 2005.

[16] Tashaphyne. Arabic light stemmer, 0.2. 2010, 2010. https://pypi.python.org/pypi/Tashaphyne/.

[17] A. Al-Subaihin, H. S. Al-Khalifa, and A. S. Al-Salman. 2011. A Proposed Sentiment Analysis Tool for Modern Arabic Using Human-Based Computing. In Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services, iiWAS '11, pp. 543–546, New York, NY, USA. ACM.

[18] J. Carletta..Assessing agreement on classification tasks: The kappa statistic.Comput. Linguist., 22(2), pp. 249–254, Jun. 1996.

[19] M. Diab. Second-generation tools (AMIRA 2.0): Fast and robust tokenization, pos tagging, and base phrase chunking.In Proceedings of the Second International Conference on Arabic Language Resources and Tools, K. Choukri and B. Maegaard, Eds. Cairo, Egypt: The MEDAR Consortium, April 2009, pp. 285–288.

[20] A. El-Khair. Effects of stop words elimination for Arabic information retrieval: A comparative study.International Journal of Computing & Information Sciences, 4(3), pp. 119–133, 2006.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825– 2830.