Combining Lexical Features and a Supervised Learning Approach for Arabic Sentiment Analysis

Conference Paper · April 2016

CITATIONS
4

READS
556

4 authors, including:

Samhaa R. El-Beltagy
Nile University
105 PUBLICATIONS **717** CITATIONS
SEE PROFILE

Muhammad Hammad
Cairo University
3 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Combining Lexical Features and a Supervised Learning Approach for Arabic Sentiment Analysis

Samhaa R. El-Beltagy, Talaat Khalil, Amal Halaby, and Muhammad Hammad

Center of Informatics Sciences, Nile University, Giza, Egypt
samhaa@computer.org, t.maher@nu.edu.eg, am.mahmoud@nu.edu.eg,
mhammad@sci.cu.edu.eg

Abstract. The importance of building sentiment analysis tools for Arabic social media has been recognized during the past couple of years, especially with the rapid increase in the number of Arabic social media users. One of the main difficulties in tackling this problem is that text within social media is mostly colloquial, with many dialects being used within social media platforms. In this paper, we present a set of features that were integrated with a machine learning based sentiment analysis model and applied on Egyptian, Saudi, Levantine, and MSA Arabic social media datasets. Many of the proposed features were derived through the use of an Arabic Sentiment Lexicon. The model also presents emoticon based features, as well as input text related features such as the number of segments within the text, the length of the text, whether the text ends with a question mark or not, etc. We show that the presented features have resulted in an increased accuracy across six of the seven datasets we've experimented with and which are all benchmarked. Since the developed model outperforms all existing Arabic sentiment analysis systems that have publicly available datasets, we can state that this model presents state-of-the-art in Arabic sentiment analysis.

1 Introduction

Social media networks are playing an increasingly important role in the transmission of opinions about almost everything. Movies, products, actors, politicians, and events, are but a few examples of entities being targeted by opinionated posts. Because of this, social media has lately turned into a decision making tool, where decisions taken can vary from which political candidate to vote for, to which product to buy or which movie to watch. As a result, sentiment analysis has been the focus of many research studies in the past few years with sentiment Analysis in Arabic following the trend. Compared to the English language, the Arabic Language remains under-resourced with respect to annotated datasets and lexicons. However, more and more Arabic resources are starting to appear.

This paper presents a model for carrying out Arabic sentiment analysis by augmenting a machine learning approach with a set of features derived from an Arabic sentiment lexicon as well as from the text itself. To validate the model, it was applied

A. Gelbukh(Ed.): CICLing,2016, Part I, LNCS 7816, pp. 89-97, 2013.

to all benchmarked datasets that the authors were able to acquire. The results of experimenting with these datasets, show that with the exception of one dataset, the model achieves higher polarity detection accuracy than all similar systems that have experimented with the same datasets.

The rest of this paper is organized as follows: section 2 briefly describes related work, section 3 overviews the proposed model and its preprocessing and feature extraction steps, section 4 presents the experiments carried out to evaluate the presented model and their results, and finally section 5 concludes this paper and presents future research directions.

2 Related Work

Work on Arabic Sentiment analysis has been gaining a lot of attention during the past couple of years. In [1], Abdulla et al, compared machine learning and lexicon based techniques for Arabic sentiment analysis on tweets written in the Jordanian dialect. The data set that was used for comparison consisted of 2000 tweets (1000 positive and 1000 negative). The preprocessing steps applied on the dataset included spelling correction, elongation and stop-word removal, and letter normalization. The authors experimented with a set of classifiers including: Support Vector Machines (SVM), Naive Bayes (NB), and K-Nearest Neighbor (KNN) with K=9, using RapidMiner [2]. The best results were reported to be those of SVM and NB (using 5-fold cross validation and light stemming) with accuracies of 87.2% and 81.3% respectively.

The work presented in [3] targeted tweets written in the Egyptian dialect and was focused on examining the effect of different pre-processing steps on the task of sentiment analysis. The tweets that were used for training were chosen such that they contained only one opinion (positive or negative), were not sarcastic, and covered different topics. The tweet training set consisted of one thousand tweets (500 positive and 500 negative) manually annotated by two experts. Preprocessing included removing user-names, pictures, hash tags, stopwords, URLs, and all non-Arabic words. 10-fold cross validation was used for evaluating system performance. SVM was used for classification, and Weka [4] was used as the platform for experimentation. The results reported on that work were on raw data, normalized data, and light stemmed data using a modified version of El-Beltagy_Rafea Stemmer [5]. The best results were obtained by applying normalization, stemming, a combination of Unigrams and Bigrams, and stop word removal.

Salamah and Elkhlifi [6] developed a system for extracting sentiment from the Kuwaiti-Dialect. The system consisted of four components: a tweets collector, a preprocessing module, an opinionated terms extractor, and an opinion classifier. The preprocessing module consisted of a segmenter developed by the authors as well as the Stanford Arabic Tokenizer and was used to extract features from each tweet. The authors implemented their own set of resources of adjectival, nominal, verbal and adverbial indicators for the Kuwaiti dialect. They experimented with a manually annotated dataset comprised of 340,000 tweets, using SVM, J48, ADTREE, and Ran-

dom Tree classifiers. The best result was obtained using SVM with a precision and a recall of 76% and 61% respectively.

In [7], Duwairi et al present a sentiment analysis tool for Jordanian Arabic tweets. The authors created three lexicons to enhance the overall system accuracy. The first lexicon was used to map all dialect words to Modern Standard Arabic (MSA), the second lexicon was used to convert all Arabizi (Arabic words written in Roman Alphabet) words to MSA, and the third lexicon was used to convert emoticons to their respective meaning in the language. One thousand manually annotated Arabic Jordanian tweets were used for evaluating system performance. Data preprocessing steps involved: tokenization, stop words removal, links and elongation removal, letter normalization and stemming. RapidMiner [2] was used to train and test the model using NB, SVM and KNN. The NB classifier performed best in this experiment with an accuracy of 76% using 5-fold cross validation without stemming or stop words removal.

The goal of the work presented by Salameh et al [8] was to investigate whether sentiment information was lost or persevered when translating from one language to another. In this study, the authors have used Arabic as a source language and English as a target and experimented with different configurations relating to how translation and sentiment annotation are carried out. In order to carry out their study, the authors adapted the NRC-Canada sentiment analysis system[9][10], which can be considered as state of the art in English sentiment analysis[1], to work with Arabic. As such, and based on the results reported in their work, the system that they have developed for Arabic can be thought of as the state of the art in Arabic sentiment analysis. Tweet pre-processing included URL and mention normalization as well as character normalization, tokenization, part of speech tagging, and lemmatization. The main features used by the system were word and character n-grams, the count of each part-of-speech tag, the number of negated contexts, and a sentiment score. The sentiment score was calculated with the aid of a sentiment lexicon. The Arabic lexicon used in this work, was an automatically generated one. Evaluation of the Arabic system was carried out by testing it on two existing datasets: Mourad and Darwish[11] and Refaee and Rieser [12] as well as two other datasets that the authors have annotated (BBN and Syria). All four datasets have been used to test the work described in this paper and are described in detail in section 4.1.

Shoukry and Rafea [13] present an approach that combines sentiment scores obtained using a lexicon with a machine learning approach and they apply it on Egyptian tweets. Pre-processing in their system involves character normalization, stemming, and stop word removal. Features are represented as a count vector of unigrams, bigrams, and tri-grams. The sentiment lexicon employed in this work, is one that was manually built by the authors. The lexicon contains 390 negative entries and 262 positive entries. To test their system, the authors annotated 4800 tweets[2] as positive, negative, or neutral (1600 positive, 1600 negative, 1600 neutral). The experiments con-

---

[1] The system was the best performer in SemEval 2013 and SemEval 2014 with respect to the message level polarity detection task [9][10]

[2] The version provided to us by the authors had 4820 tweets.

ducted by the authors, show that adding the semantic orientation feature does in fact improve the result of the sentiment analysis task.

3 Model Overview

3.1 Model components

The model that we have adopted is one that employs statistical machine learning for evaluating the polarity of some input text. Our previous work presented in [14], has revealed that classifiers that perform best for the task of sentiment analysis are those that belong to the family of Naïve Bayes as well as SVM, but that the best performer amongst those, varies from one dataset to another. This conclusion is supported by the literature as presented in section 2, where some authors have reported that they obtained their best results using an SVM classifier while others reported that Naïve Bayes classifiers performed better than SVM. In our model, we have chosen to use Complement Naïve Bayes (CNB)[15] as the default classifier. This choice is also based on experiments reported in [14], where CNB performed consistently well across experimented with datasets, even if it was not always the top performing classifier across these datasets. However, our system has been built in such a way that we can easily interchange CNB with any other classifier. The actual implementation of the system, was carried out in Java, with the Weka library [16] providing the machine learning functionality.

Like all supervised sentiment analysis systems, the first step to be carried out in our model is data pre-processing followed by feature extraction. Features extracted in the latter step, represent the actual inputs to the target classifier. The main contribution of the presented work is the introduction of the set of features that are extracted from input text and which we argue, improve classification accuracy. While some of these features are related to input text characteristics, such as the length of the text, whether it ends with a question mark or not, etc, other features are related to the occurrence of sentiment words or phrases within the input text. To extract these features a sentiment lexicon is needed. For the past few years, Arabic has been considered as an under-resourced language with respect to the task of sentiment analysis due to the almost non-existence of sentiment lexicons and training datasets annotated with sentiment. As time goes by, more and more annotated sentiment public datasets are starting to emerge [17][12][8]. Quality lexicons are still scarce, although efforts have been made to translate existing English lexicons to Arabic in order to fill this gap [11][8]. To our knowledge, the largest Arabic sentiment lexicon with a manual like quality is NileULex which is presented in [18]. The lexicon has 5953 positive and negative entries and includes sentiments words and phrases from both Modern Standard Arabic and Egyptian. Many of the dialectical Egyptian entries, are also used in other Arabic speaking countries. In the presented work, we make use of this lexicon. The following two subsections, detail the pre-processing steps carried out in our model, and the features that are extracted and used by the sentiment classifier.

3.2 Preprocessing Steps

In this section, we present a detailed description of the preprocessing pipeline for our sentiment classifier.

1. Character Normalization

The first preprocessing step applied to input text, is character normalization. In this step, letters "أ" , "إ" and "آ" are replaced with "ا" while the letter "ة" is replaced with "ه", and the letter "ى" is replaced with "ي". Diacritics are also removed in this step.

2. Elongation Removal

In this step, words that have been elongated, are reduced to their normal standard canonical form. In social media, elongation is a method for giving certain words more emphasis. An example of an elongated word is "رااااااااااائع" (magnificent). An English example is "yesssssss". The algorithm applied for elongation detection and removal is a simple one. A regular expression is used to detect if a character appears consecutively three or more times. If such a character is detected, the consecutive repetitions are replaced by a single instance of that character. As a result "yesssssss" will be transformed to "yes" and "رااااااااااائع" to "رائع".

3. Emoticons Detection and Replacement

In this step, input text is matched against a predefined list of emoticons labeled as positive or negative. For this step, we have compiled a list comprised of 105 negative emoticons and 110 positive emoticons. All matched emoticons are replaced with a single term that is not actually part of the Arabic vocabulary depending on whether the match is with a positive or a negative emoticon. In the actual implementation of our system, any positive emoticon is replaced by the term "ايموشنموجب" while negative emoticons are replaced by the term"ايموشنسالب". The number of emoticon matches encountered is stored as they it considered as part of the feature set.

4. Mention Normalization

In this step, any mention starting with @, is replaced with the English word "MENTION".

5. Named Entities Tagging

Although Named Entities identification may not seem to be directly related to sentiment analysis, as detailed in [19], in the absence of POS tags, many Arabic names can get confused with sentiment lexicon entries, which can have a negative impact over the overall accuracy of a sentiment analysis system. For example, without a named entity recognition (NER) system the word "طيبه" in the term "جامعه طيبه" (Teaba University), will match with lexicon entry "طيبه" which means "kind" or "kindness" depending on the context. There are many other examples given in in [19]. We have used the system described in [20] to tag named entities. Through experimentation, we discovered that the effect of NER is not as great as we expected it to be, but that its inclusion still has a positive effect.

6. Matching with Lexicon Entries

In this step, input tweets/texts are matched against entries in the sentiment lexicon. Since Arabic is a morphologically complex language, having a lexicon containing all

possible surface forms of its entries is an almost impossible task[8][11]. On the other hand, straight forward stemming, even if just light, can alter the meaning of a word entirely. If we take for example the word "روعه" which means "magnificent", and stem it using any traditional Arabic light stemmer, the result will be the term "روع", which means "terrorized". While the first term is very positive, the second is very negative. This problem can be avoided through the use of lemmatization instead of stemming. To carry out lemmatization we have used a dictionary based tool [21] which is based on the work presented in [5]. The tool utilizes a large set of dictionaries built using large datasets as described in [5] and [22]. In addition, the tool allows for the addition of manual entries. Base forms of lexicon entries were added to a "stem list"[5] to ensure that any word matching with those, does not get stemmed beyond its base form. Both the tweets/texts and lexicon entries are lemmatized and stemmed using this tool, prior to any matching steps. When a match occurs between text in the input, and a lexicon entry, a unique term is added to the input text immediately after the matching sentiment term depending on whether the match was with a positive or negative entry (we are using the English terms "pos" and "neg" as sentiment identifiers). Example:

النهايات **السعيده** pos دائما تاتي **في المشمش**neg

In the shown example, terms that matched with a lexicon entry are underlined and in bold. A count for positive and negative terms is also kept to be later used as part of the features. Negators are currently handled in a very simple way: encountering a negator before a sentiment term within a window ***w*** results in the reversal of its polarity. We have observed that in some cases, this is not necessarily valid. For example, the term "لا حلو", in which the negator "no" appears before the word "nice", is actually used to affirm that something is nice.

7. Stemming

For stemming, we used the same tool described in step 6. However, here we used it in "stem enforce" mode. In this mode, any ta'a marbota (ة), ha'a (ه) or a trailing ya'a (ي) are removed even if they are part of a word that exists in the stemmer's dictionaries. However, other terms that are in the dictionaries and that have known suffixes or prefixes will be preserved. For example, the word حيوان will not be stemmed. The tool is also capable of reducing many broken plurals to their singular form.

3.3 Features

The following is the list of features used by our model:

- Stemmed word uni-grams and bi-grams represented by their idf weights. (Terms whose occurrence count is less than 2 are excluded from the feature vector).
- *startsWithLink*: a feature which is set to 1 if the input text starts with a link and to a 0 otherwise.
- *endsWithLink*: a feature which set to 1 if the input text ends with a link and to 0 otherwise.
- *numOfPos*: a count of terms within the input text that have matched with positive entries in the sentiment lexicon. To give some extra weight to terms that are made up of more than one word (compound_terms) the following formula is used to set this feature:

$$numOfPos = \sum_{i=0}^{n} i + \sum_{j=0}^{c} j \times \alpha \quad (1)$$

Where ***n*** is the number of positive single terms, ***c*** is the number of positive compound terms, and α is the boosting factor >1. We have set α to 1.5 based on empirical experimentation.

- *numOfNeg*: a count of terms within the input text that have matched with negative entries in the sentiment lexicon. The value of this feature is calculated in the same way the *numOfPos* is calculated.
- *length*: a feature that can take on one of three values {0,1,2} depending on the length of the input text. The numbers correspond to very short, short and normal. A tweet is categorized as "very short" if its length is less than 60 characters, "short" if it is less than 100, and normal otherwise.
- segments: a count for the number of distinct segments within the input text. We assume that segments are delimited by any of the following characters: "-|?.،؛ ؟,!:"
- *endsWithPostive*: a flag that indicates whether the last encountered sentiment word was a positive one or not.
- *endsWithNegative*: a flag that indicates whether the last encountered sentiment word was a negative one or not.
- *negPercentage*: a real number from 0 to 1 that represents the percentage of words in the text that are negative. For example, given the text " حسبي الله ونعم الوكيل", this number will be set to 1, as the entire text appears as a single entry in the lexicon. Given the text "ياه، ده اوي **وحش**", this number will be set to 0.25 as 1 word out of the four in the text is negative.
- *posPercentage*: a real number from 0 to 1 that represents the percentage of words in the text that are positive.
- *startsWithHashTag*: a flag that indicates whether the tweet starts with a hashtag.
- *numOfNegEmo*: the number of negative emoticons that have appeared in the tweet.
- *numOfPosEmo*: the number of positive emoticons that have appeared in the tweet.
- *endsWithQuestionMark*: a flag that indicates whether the tweets ends with a question mark or not.

4 Experiments and Results

The goal of the presented experiments was to determine whether the features introduced by this work do in fact improve sentiment analysis results or not. To determine this, we needed to compare between our model and exiting benchmarks. Towards this end, we've collected as many benchmarked datasets as we were able to get. The description of the used datasets is provided in section 4.1. The previously obtained results for these datasets can be found in [11], [8] and [14]. Description of the experiments and their results, can be found in section 4.2. All experiments were carried out using the WEKA workbench [4].

4.1 The used Datasets

The Talaat et al dataset (NU) [14]: The collection and annotation for this dataset is described in [14]. The dataset contains 3436 unique tweets, mostly written in Egyptian dialect. These tweets are divided into a training set consisting of 2746 tweets and a test set containing 683 tweets. The distribution of training tweets amongst polarity classes is: 1046 positive, 976 negative, and 724 neutral tweets. The distribution of the test dataset is: 263 positive, 228 negative and 192 neutral. This dataset is available by request from the author of this paper.

The KSA_CSS dataset (NBI) [14]**:** this dataset is one that was collected at a research center in Saudi Arabia under the supervision of Dr. Nasser Al-Biqami and which is also described in [14]. The majority of tweets in this dataset are in Saudi and MSA, but a few are written in Egyptian and other dialects. The tweets for this dataset have also been divided into a training set consisting of 9656 tweets and a test set comprised of 1414 tweets. The training set consists of 2686 positive, 3225 negative, and 3745 neutral tweets and the test set has 403 positive, 367 negative, and 644 neutral tweets.

The Refaee_Rieser Dataset (RR2, RR3) [12]**:** This dataset is a twitter subset of a dataset collected by Refaee & Rieser [23]. This particular dataset does not target a specific dialect. We re-constructed the dataset using the Twitter IDs provided by the authors, but some of those were deleted or not found. As a result the re-constructed dataset consisted of 724 positive tweets, 1565 negative tweets, and 3204 neutral tweets. In order to compare our model with existing benchmarks, this dataset was used in two configurations. In the first it was divided into a training set (4405 tweets) and a test set (1088 tweets) as described in [14]. The training dataset had 599 positive, 1241 negative, and 2565 neutral tweets, and the test dataset had 125 positive, 324 negative, and 639 neutral tweets. In the second configuration, the neutral class was omitted leaving 2289 tweets divided into 1563 negative tweets and 722 positive tweets.

The Mourad_Darwish Dataset (MD) [11]: Like the the Refaee_Rieser Dataset, this dataset does not target a specific dialect. The dataset has a total of 1111 tweets of which 377 are classified as negative and 734 are classified as positive.

The BBN Dataset (BBN) [8]: This dataset consists of 1199 Levantine sentences, selected by the authors of [8] from LDC's BBN Arabic-Dialect–English Parallel Text. The sentences were extracted from social media posts. The polarity breakdown of the sentences in this dataset is as follows: 498 are positive, 575 are negative, and 126 are neutral.

The Syria Dataset (SYR) [8]: This dataset consists of 2000 Syrian tweets, so most of the tweets in this dataset are in Levantine. The dataset was collected by (Salameh and Mohammad) [8] and consists of 448 positive tweets, 1350 negative tweets and 202 neutral tweets.

The Shoukry_Rafea Dataset (SR) [13]**:** This dataset consists of 4820 Egyptian tweets divided into 1604 negative tweets, 1612 positive tweets and 1604 neutral tweets. The authors of this dataset have kindly shared it with us, but the shared version is one that has been pre-processed by removing all mentions, hashtags and URLs.

Table 1 provides presents an overview of each of the used datasets described above.

Table 1. Summary of the size and distribution of used datasets among polairty classes

		Number of Tweets							
Dataset	***Total***	***Training***				***Testing***			
		Pos	***Neg***	***Neu***	***Total***	***Pos***	***Neg***	***Neu***	***Total***
NU	3436	1046 (38.1%)	976 (35.5%)	724 (26.4%)	2746	263 (38.5%)	228 (33.4%)	192 (28.1%)	683
NBI	11070	2686 (28.1%)	3225 (33.7%)	3745 (39.2%)	9566	403 (28.5%)	367 (26.0%)	644 (45.5%)	1414
RR2	2285	722 (31.6%)	1563 (68.4%)	-	2285	-	-	-	-
RR3	5493	599 (13.6%)	1241 (28.2%)	2565 (58.2%)	4405	125 (11.5%)	324 (29.8%)	639 (58.7%)	1088
MD	1111	734 (66.1%)	377 (33.9%)	-	1111	-	-	-	-
BBN	1199	498 (41.5%)	575 (48%)	126 (10.5%)	1199	-	-	-	-
Syria	2000	448 (22.4%)	1350 (67.5%)	202 (10.1%)	2000	-	-	-	-
SR	4820	1612 (33.4%)	1604 (33.3%)	1604 (33.3%)	4820	-	-	-	-

4.2 Results

We tested our model which relies on the features presented in section 3.2, on each of the datasets described in the previous subsection. Table 2, shows the accuracy results of applying 10 fold cross validation on the MD, RR2, BBN, SYR, and SR datasets. Each of these datasets has at least one previously published result as indicated in the table. Results published in Salameh et al[8], were provided in two contexts that are relevant to this work. In the first (Ar Sys), polarity annotations were performed directly on Arabic text and were compared to system generated annotations. In the second (Eng Sys), automatically translated Arabic text was classified using an English sentiment analysis system and the labels were compared against original Arabic labels for the same text.

The results for our system are shown in two formats. The first format (default configuration) employs a Complement Naïve Bayes (CNB) classifier [24] with a smoothing factor of 1.0. Practical experience has shown us that smaller CNB smoothing factors result in better cross validation results, but poor test results.

Table 2. Comparison between the accuracy (%) of various sentiment analysis systems on some of the used datasets

Dataset	**MD**	**RR2**	**BBN**	**SYR**	**SR**
labels	pos,neg	pos,neg	pos,neg,neu	pos,neg,neu	pos,neg,neu
size	1111	2285[3]	1199	2000	4820
Baseline	67.87	66.78	56.88	75.4	73.84
Mourad & Darwish [11]	72.5	_	_	_	_
Salameh et al[8] (Ar Sys)	74.62	85.23	63.89	78.65	_
Salameh et al (Eng Sys)	_	_	62.49	78.11	_
Shoukry & Rafea	_	_	_	_	80.6
Our System (default config)	80.2	84.55[4]	**71.06**	78.75	83.03[5]
Our System (Best Classifier)	**81** (MNBU)	**85.03** (SVM)	**71.06** (CNB)	**80.6** (SVM)	**83.13** (MNBU)

As previously stated, work presented in [14] has shown the best classifier to use for some given dataset, is often dataset dependent with Naïve Bayes classifiers and SVM usually yielding the best results. Accordingly, we have also included another format for our model, where the best results obtained by one those classifiers is outlined. MNBU in the table, refers to a multi-nominal updateable Naïve Bayes classifier[25]. The baseline in this table, was obtained by representing raw unprocessed tweets using boolean vectors of unprocessed unigrams and classifying them using SVM (Linear Kernel with LibSVM's [26] default parameters).

The results shown in table 2, illustrate that even when using the default configuration (which is not always the best), the presented model outperforms all other existing systems on all datasets except for RR2. The version of the RR2 dataset that we are using is smaller than that on which results in the table have been presented, so the results are not directly comparable. We've also found that using information gain (IG) to reduce the features for this dataset as well as for the SR dataset, improves the overall accuracy. The result for the SR dataset for example, would drop from 83.03% to 81.08% without using IG.

Table 3 shows the results of applying our model on datasets presented in [14]. In this table, the best results obtained in[14] are reported although it was often the case in this study that classifiers that scored best for 10 fold cross validation were not the ones that scored best for the test datasets. The baseline in this experiment has been calculated in the same way as for the experiment presented in table 2. From the results it can be seen that the presented model performs quite well with respect to both the

[3] The version of the dataset that we have used is smaller than that used by[8], so our results and theirs are not directly comparable.

[4] In our system, features for this dataset were reduced using information gain

[5] Features were reduced for this dataset using information gain

NU and NBI datasets. However, it performs rather badly with the RR3 dataset. When using CNB, which is the default classifier, the accuracy even drops below the baseline. We are not sure why the system performance drops like this with this particular dataset, but this will be subject to further analysis.

Table 3. Comparison between proposed model results and results presented in [14]

Dataset	NU		NBI		RR3	
	10 FCV	**Test**	**10 FCV**	**Test**	**10 FCV**	**Test**
Size	2746	683	9656	1414	4405	1088
Baseline	55.86	54.76	66.73	66.62	53.86	54.56
Talaat et al [14]	70.84	57.25	77.34	69.81	66.7	**59.45**
Our System (default config)	**73.53**	59.23	79.06	**75.05**	63.8	53.7
Our System (Best Classifier)	**73.53** (CNB)	**60.76** (MNBU)	79.06 (CNB)	**75.05** (CNB)	**67.3** (SVM-RBF)	58.8 (SVM-RBF)

5 Conclusion and Future Work

In this paper we have presented a set of features that can be used with Arabic tweets in order to enhance the performance of a sentiment analysis system. We believe that the features that have the highest impact in enhancing the results, are those derived from a high quality sentiment lexicon. This hypothesis is supported by the work presented in [18]. By using these features within a sentiment analysis system, we have shown that our model outperforms all existing sentiment analysis systems on 6 out of the 7 datasets on which we have applied it.

In the future we would like to enhance the presented model by: handling negation in a better way, using elongation as an indicator for emphasis, and making use of intensifiers. We would also like to assign weights to various entries in the lexicon and use scores instead of counts. We have already taken initial steps towards this goal which are presented in [27]. We would also like to investigate augmenting the currently used lexicon with some or all of the currently available translated lexicons.

6 Acknowledgements

The authors would like to thank Amira Shoukry, Dr. Ahmed Rafea, and Dr. Kareem Darwish for kindly sharing their datasets.

7 References

1. Abdulla, N.A., Ahmed, N.A., Shehab, M.A., Al-ayyoub, M.: Arabic sentiment analysis: Lexicon-based and corpus-based. In: Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on. pp. 1–6. IEEE, Amman (2013).
2. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid prototyping for complex data mining tasks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 935–940 (2006).
3. Shoukry, A., Rafea, A.: Preprocessing Egyptian Dialect Tweets for Sentiment Mining. In: Proceedings of the fourth workshop on Computational Approaches to Arabic Script-Based Languages. pp. 47–56. , San Diego, California, USA (2012).
4. Witten, I.H., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.J.: Weka : Practical Machine Learning Tools and Techniques with Java Implementations. Seminar. 99, 192–196 (1999).
5. El-Beltagy, S.R., Rafea, A.: An Accuracy Enhanced Light Stemmer for Arabic Text. ACM Trans. Speech Lang. Process. 7, 2 – 23 (2011).
6. Salamah, J. Ben, Elkhlifi, A.: Microblogging Opinion Mining Approach for Kuwaiti Dialect. In: The International Conference on Computing Technology and Information Management (ICCTIM2014). pp. 388–396 (2014).
7. Duwairi, R., Marji, R., Sha'ban, N., Rushaidat, S.: Sentiment Analysis in Arabic tweets. In: Information and Communication Systems (ICICS), 2014 5th International Conference. pp. 1 – 6. IEEE, Irbid (2014).
8. Salameh, M., Mohammad, S., Kiritchenko, S.: Sentiment after Translation: A Case-Study on Arabic Social Media Posts. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 767–777. Association for Computational Linguistics, Denver, Colorado (2015).
9. Mohammad, S.M., Kiritchenko, S., Zhu, X.: NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In: Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013). , Atlanta, Georgia, USA (2013).
10. Kiritchenko, S., Zhu, X., Mohammad, S.: Sentiment Analysis of Short Informal Texts. J. Artif. Intell. Res. 50, 723–762 (2014).
11. Mourad, A., Darwish, K.: Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs. Proc. 4th Work. Comput. Approaches to Subj. Sentim. Soc. Media Anal. 55–64 (2013).
12. Refaee, E., Rieser, V.: Subjectivity and Sentiment Analysis of Arabic Twitter Feeds with Limited Resources. In: Proceedings of the Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools. pp. 16–21. , Reykjavik, Iceland (2014).
13. Shoukry, A., Rafea, A.: A Hybrid Approach for Sentiment Classification of Egyptian Dialect Tweets. In: First International Conference on Arabic Computational Linguistics (ACLing). pp. 78–85. , Cairo, Egypt (2015).
14. Khalil, T., Halaby, A., Hammad, M.H., El-Beltagy, S.R.: Which configuration works best? An experimental study on Supervised Arabic Twitter Sentiment Analysis. In: Proceedings of the First Conference on Arabic Computational Liguistics (ACLing 2015), co-located with CICLing 2015. pp. 86–93. , Cairo, Egypt (2015).
15. Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.R.: Tackling the Poor Assumptions of Naive Bayes Text Classifiers. Proc. Twent. Int. Conf. Mach. Learn. 20, 616–623 (2003).
16. Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H., Trigg, L.: WEKA: A Machine Learning Workbench for Data Mining. In: Maimon, O. and Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook. pp. 1305–14. Springer (2005).

17. ElSahar, H., El-Beltagy, S.R.: Building Large Arabic Multi-domain Resources for Sentiment Analysis. In: Proceedings of CICLing 2015, Volume 9042 of the series Lecture Notes in Computer Science. pp. 23–34. Springer Verlag (2015).
18. El-Beltagy, S.R.: NileULex: A Phrase and Word Level Sentiment Lexicon for Egyptian and Modern Standard Arabic. In: to appear in proceedings of LREC 2016. , Portorož , Slovenia (2016).
19. El-Beltagy, S.R., Ali, A.: Open Issues in the Sentiment Analysis of Arabic Social Media : A Case Study. In: Proceedings of 9th the International Conference on Innovations and Information Technology (IIT2013). , Al Ain, UAE (2013).
20. Omnia Zayed, Samhaa R. El-Beltagy: Named Entity Recognition of Persons' Names in Arabic Tweets. In: Proceedings of Recent Advances in Natural Language Processing (RANLP 2015). , Hissar, Bulgaria (2015).
21. El-Beltagy, S.R., Rafea, A.: LemaLight: A Dictionary based Arabic Lemmatizer and Stemmer. (2016).
22. El-Beltagy, S.R., Rafea, A.: A corpus based approach for the automatic creation of arabic broken plural dictionaries. In: Computational Linguistics and Intelligent Text Processing. pp. 89–97. Springer Berlin Heidelberg (2013).
23. Refaee, E., Rieser, V.: An Arabic Twitter Corpus for Subjectivity and Sentiment Analysis. In: Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC'2014). , Iceland (2014).
24. Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.R.: Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In: Proceedings of the 20th International Conference on Machine Learning, (ICML '03). pp. pp. 616–623. , USA (2003).
25. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI/ICML-98 Workshop on Learning for Text Categorization. pp. 41–48 (1998).
26. Chang, C., Lin, C.: LIBSVM : A Library for Support Vector Machines. ACM Trans. Intell. Syst. Technol. 2, 1–39 (2011).
27. El-Bletagy, S.R.: NileTMRG: Deriving Prior Polarities for Arabic Sentiment Terms. In: Proceedings of SemEval 2016 -(submitted). , San Diego, California (2014).