Journal of Mobile Multimedia, Vol. 13, No.3&4 (2017) 233-243
© Rinton Press

SENTIMENT CLASSIFICATION OF ARABIC TWEETS: A SUPERVISED APPROACH

NAAIMA BOUDAD
ENSIAS, Mohammed V University, Rabat
naaima.boudad@gmail.com

RDOUAN FAIZI
ENSIAS, Mohammed V University, Rabat
rdfaizi@gmail.com

RACHID OULAD HAJ THAMI
ENSIAS, Mohammed V University, Rabat
rachid.ouladhajthami@gmail.com

RADDOUANE CHIHEB
ENSIAS, Mohammed V University, Rabat
radchiheb@gmail.com

Social media platforms have proven to be a powerful source of opinion sharing. Thus, mining and analyzing these opinions has an important role in decision-making and product benchmarking. However, the manual processing of the huge amount of content that these web-based applications host is an arduous task. This has led to the emergence of a new field of research known as Sentiment Analysis. In this respect, our objective in this work is to investigate sentiment classification in Arabic tweets using machine learning. Three classifiers namely Naïve Bayes, Support Vector Machine and K-Nearest Neighbor were evaluated on an in-house developed dataset using different features. A comparison of these classifiers has revealed that Support Vector Machine outperforms others classifiers and achieves a 78% accuracy rate.

Keywords: Sentiment Analysis; opinion mining; Arabic; Twitter; Machine Learning; Supervised Approach

1. Introduction

With the emergence of social network services, the internet user has become a major player who interacts, collaborates with others and gives his opinion on different issues and products. The frequent use of these services provides a high-value knowledge source for business companies to measure customer satisfaction and monitor their competitive environment. Therefore, opinion mining has drawn the attention of many researchers.

Sentiment Analysis, also called Opinion Mining, uses natural language processing, text analysis and computational linguistics techniques to identify and extract sentiments expressed in a given text. One major task in this domain is sentiment classification, which aims to determine whether the semantic orientation of a text is positive, negative or neutral.

Sentiment Analysis can be investigated at three levels of granularity, namely document level, sentence level, and aspect level. At the document level, the whole piece of writing is dealt with as one unit and is assigned to positive, negative or neutral classes. This level of analysis supposes that each document expresses an opinion on a single entity and has only one opinion holder. At the sentence level, Sentiment Analysis aims to identify whether the sentence holds an opinion or not, and evaluates the sentiment orientation of sentences. This level of granularity is more challenging by the fact that the sentiment orientation of words is highly context-dependent. By contrast, the aspect level performs finer-grained analysis, and its purpose is to identify the aspects of the opinion target and the sentiment expressed towards each aspect. A positive opinion on an object can be positive on just an attribute of the object, but not on the object as a whole [1].

Two types of methods have generally been used in the literature to tackle document sentiment classification: Machine learning and lexicon-based methods. The first method is considered as a text classification problem, which makes use of machine learning algorithms and linguistic features. Lexicon-based methods use of a set of words associated with their semantic orientation to identify the overall class of the inputted text.

In this study, analysis is confined to sentiment classification of Arabic tweets. In fact, Twitter has become a very popular microblogging platform in the Arab world on which users share opinions on various topics. Arabic tweets are automatically collected and labelled to train machines learning classifiers such as Naive Bayes and Maximum entropy.

The rest of the paper is organized as follows. In Section 2, we review a selection of studies on Arabic sentiment classification. In Section 3, we present our data set and detail our proposed approach. Experimental results and evaluation are presented in Section 4. Finally, we conclude in Section 5.

2. Related Work

In their study of sentiment classification in Arabic, Mountassir et al. [2] used three classifiers: NB, SVM and KNN. In addition, they made use of two corpora. The first corpus, combining two domain-specific datasets (namely, movies and sports), was collected by the authors themselves. The second is OCA, a corpus of movie-reviews developed by Rushdi-Saleh et al. [3]. Before the classification stage, the authors carried out a pre-processing task by removing all the stop words, separating the words from their clitics, discarding the entities that are used only once or twice in the dataset, and by substituting the words with their stems. The findings of their study demonstrated that pre-processing, n-grams combining, and presence-based weighting enhance the classification performance.

In [4], two different forms of sentiment classification were investigated: the polarity classification, which categorizes reviews as having either a positive or a negative sentiment, and the rating classification, which rates a given review on a scale of 1 to 5. Aly and Atiya [4] developed a Large-scale Arabic Book Review (LABR), a dataset that is composed of more than 63,257 book reviews collected from www.goodreads.com. Reviews whose rating is either 4 or 5 were noted to be positive. By contrast, reviews which were assigned 1 or 2 were labelled negative. Reviews that were rated 3 were classified as neutral. Given that the number of positive reviews (42,832) was higher than that of negative reviews (8224), the authors resorted to machine learning by using SVM, MNB and BNB as algorithms and n-grams as features. As far as sentiment polarity classification is concerned, the assessment of their dataset achieved quite good results (~90% accuracy). Yet, the rating classification needs much more improvement (~50% accuracy).

In their contribution, El-Baltagy et al. [5] put forward a lexicon-based approach to classify sentiments in Egyptian texts. After setting up a lexicon of 4392 words, the authors used two distinct datasets (i.e. a Twitter dataset composed of 500 tweets and Dostour dataset that includes 100 comments from the Web) so as to evaluate two unsupervised classification algorithms. The first algorithm computes one score for every document by adding up the weights of positive and negative terms. The second assigns every word in the lexicon a positive and a negative weight and then calculates the positive and negative scores of each document. Results of the analysis clearly proved that the use of both algorithms on a Twitter dataset achieved good results (83.8 % accuracy).

For their parts, El-Makky et al. [6] put forward a hybrid approach based on Sentiment Orientation algorithms together with a machine learning classifier. For every document in a twitter dataset, the authors used the lexicon-based approach to calculate Sentiment Orientation scores. The latter were added a variety of features namely, unigrams, language independent attributes, Tweets-specific and stem polarity features so as to create an input feature vector for the SVM classifier. This dual use of both the Machine Learning classification approach and the lexicon based model yielded somewhat better results than one single approach results (accuracy 84%).

In their research work on sentiment intensity in Arabic phrases, Kiritchenko et al. [7] noted that the objective of task 7 of SemEval 2016 was to provide a score (between 0 and 1) that indicates the sentiment intensity of a phrase. Score 1 refers to the maximum of "positive strength" whereas score 0 denotes the maximum of "negative strength". The participants in the study were provided with a development set of 200 terms frequently found in Arabic tweets and a set of 1166 terms for the evaluation period. Three systems were submitted by three teams: NileTMRG [8], iLab-Edinburgh [9] and LSIS [10].

Using a supervised approach, NileTMRG team [8] collected 249 K tweets by querying Twitter using test set terms. This collection was then classified by making use of the sentiment analyzer that they designed [11] with the Complement Naïve Bayes classifier [12] and trained on 11242 Arabic tweets. To assign a given word sentiment intensity, the normalized pointwise mutual information related to the positive class was calculated and re-scaled to that the values could vary from 0 to 1.

By contrast, the system that is proposed by iLab-Edinburgh team [9] uses a hybrid approach of rule-based methods and supervised learning. The system firstly uses Linear Regression trained with the labMT1.0 Sentiment Lexicon [13] that is a publicly available list of 10k Arabic positive/negative terms. Each entry of the lexicon is associated with its Sentiment Intensity score ranging between 1 and 9 (with 1 being very negative and 9 very positive). After re-scaling the Sentiment Intensity score to [0,1] and going through the training stage, the Linear Regression model is used to compute an initial Sentiment Intensity score. In the rule-based phase of the system, hand-crafted rules combined with three publicly available sentiment lexica (i.e. ArabSenti[14], MPQA[15], and Dialect lexicon [16]) were used to fine-tune initial Sentiment Intensity scores.

As for the LSIS team [10], they proposed an unsupervised method that calculates the degree of dependence between a given word and the positive class in sentiment lexica by using pointwise mutual information. If a word is not available in sentiment lexica, a web search engine was opted for to compute its sentiment orientation based on its co-occurrence near a positive or a negative word.

A thorough investigation revealed that the iLab-Edinburgh team presented the best performing system and the findings that they achieved using supervised methods are specifically higher than those got using unsupervised methods. However, the results obtained on the Arabic Twitter dataset were lower than those reached on a parallel English Twitter dataset [7].

3. Proposed Approach

In order to explore sentiment classification in the Arabic language, we propose to conduct a supervised approach on a dataset of Arabic tweets. This approach involves four consecutive tasks: data collection, text pre-processing, feature extraction, and sentiment classification.

3.1. *Data Description*

To train classifiers and evaluate our approach, we need to create a labelled dataset. Our choice of Twitter as the source for Arabic data collection is justified by four reasons. First, it is widely used as a social network in the Arab world. As such it provides a large volume of Arabic opinions on a variety of topics. Second, it is publicly available and any one can use tweets without concerns about confidentiality. Third, its limitation to 140 characters per tweet, which forces users to express their opinions in a very condensed way. Finally, it provides a free API that allows getting required tweets.

Tweets were collected by using Twitter4J, a Java library for the Twitter API. The library was configured to extract only Arabic language tweets. To generate a multi-domain dataset, we executed different search queries with multi domain keywords. The collected data was examined manually to filter duplicated tweets and to label each tweet as 'objective', 'positive', 'negative' or 'other'. The label "other" is assigned to a tweet when it expresses ambiguity or sarcasm. Since our work focuses on sentiment classification, only positive and negative tweets were kept in our database. Moreover, as the number of negative tweets (1002 tweets) widely exceeds positive ones (462 tweets), and in order to use a balanced dataset, we carried out our experiments on 996 tweets (462 positive and 534 negative).

3.2. Pre-processing

Tweets are generally noisy and unstructured, which requires some pre-processing steps before starting the classification task. The pre-processing consists of four steps:

- Normalization: In this step, we remove all the special characters, targets (@), hashtags (#), URLs and non-Arabic characters. Moreover, the Arabic letters (أ, إ, آ, ٱ) are replaced with (ا) while the letter (ة) is replaced with (ه), and the letter (ى) is replaced with (ي).

- Tokenization: Tweets are split into a sequence of tokens using all non-letter characters. This step allows us to model a text as a word vector.

- Stop word removal: This step filters Arabic stop words by removing every token equal to an item from the stop word list. We used the Arabic stop-words list obtained from the Khoja stemmer tool [6]. As this list was made for general text pre-processing, we manually inspected it to eliminate some elements that can be useful in sentiment analysis such as negation.

- Stemming: this process aims normalizing word variations by removing prefixes and suffixes and reducing words to their original root. We investigated two types of stemming. The first transforms each term to its three-letter root. The second type of stemming, called light stemming, reduces each term by removing its prefixes and suffixes without reducing them to their roots. Applying stemming algorithms reduces the number of features since many terms that are created from the same stem are represented as one feature. This technique decreases the size of document vectors and reduces the time of learning and classification processing. Table I shows some examples of stemming and light stemming.

Term		Light stemming		Stemming	
اللاعبون	the players	لاعب	player	لعب	play
ملعبهم	Their stadium	ملعب	stadium	لعب	play
اللعبة	the game	لعبة	game	لعب	play

Table 1. Stemming and light stemming

3.3. Features Extraction

After the pre-processing task, tweets were represented as a word vector. To take in consideration the order of words in the text, we added an n-gram model that presents each feature as contiguous sequence of n words [17]. A unigram is a single word, and a bigram is a couple of words that appear next to one another [18]. For example, in the sentence “لم يعجبني الفيلم” (I didn’t like the movie), the unigrams that we can found are “الفيلم“ ,”يعجبني“ ,“ لم”. While the bigrams are “لم يعجبني” and “ يعجبني الفيلم”.

To get a numerical representation of the text data we used two weighting schemes:

-TF-IDF (term frequency–inverse document frequency): calculated using tf_i the number of times a feature i occurs in a tweet and df_i the number of tweets containing the feature i.

$$\text{Weight} = tf_i * \text{Log}\,(D/df_i)$$

The tf-idf value increases uniformly to the number of times a feature occurs in the tweet, but is adjusted by the frequency of the feature in the dataset, which minimizes the weight of some words that appear more frequently in general.

-BTO (Binary-Term Occurrence): focuses on existence rather than occurrence term in a given document. BTO equal to 1 if feature appears in the tweet and 0 otherwise. This weighting was used initially in sentiment analysis by Pang et al. [19].

3.4. Learning Algorithm

Choosing an adequate learning algorithm is highly important to get good classification results. Naïve Bayes, Support Vector Machine and KNN, are the most sentiment classification algorithms used in the literature [20].

Naïve Bayes (NB):

Naïve Bayes classifier is based on a probabilistic approach that assume strong independence of features. It predicts class membership by computing the probability that a given vector belongs to a particular class. Despite its naivety, NB classifier is one of the well-performed classifiers in text classification [19].

Support Vector Machine (SVM):

Support Vector Machine classification defines a hyper-plan that divides training data points into two separate classes. The selected hyper-plane creates the largest gap between the two classes. Classification of new vectors is then based on which side of the hyper-plan they fall on. SVM has been adopted in several previous sentiment classification works and it has reported as one of the most efficient classifiers [21].

K-Nearest Neighbor (K-NN):

K-Nearest Neighbor algorithm classifies a document based on its K neighbors. These neighbors are chosen from the closest training documents in the feature space. Explicitly, it computes the distance between the unclassified document and the specified training documents. The predicted value is obtained using the majority voting or weighted average of the labels of its k nearest documents. KNN classifier has proved to be very effective in texts classification and sentiment analysis [22].

4. Experiments and discussion

4.1. Experimental setting

Our experiments aim at comparing the classification effectiveness obtained on the collected data set adopting different settings and varied classifiers. Tweets were represented in different models resulted from varying each time one of following elements:

-Stemming type: As we already mentioned, we chose to investigate the impact of two types of Arabic stemming: stemming and light-stemming.

-N-gram type: An n-gram allows us to see which words tend to occur together. It is helpful in capturing negated words. We tested uni-gram and bi-gram representation.

-Weighing type: To examine the influence of weighing scheme, we computed the weight of each feature using two different methods: TF-IDF (term frequency–inverse document frequency) and BTO (Binary-Term Occurrence).

As shown in Table 2, combining previous settings provides eight models of tweet representation.

We used SVM, K-NN and Naive Bayes methods to classify collected tweets into negative or positive class. For the K-NN classifier, we carried out a set of experiments with different values of K in order to choose the value that allows us to achieve the best results. As Fig. 1 shows, the K-NN classifier performed best when K= 9. This value is considered for the rest of our experiments.

Merging the eight vector representations and the three classifiers resulted in 24 experiments. Note that, for all representation models, normalization, tokenization and removing stop words were carried out. The next section presents the evaluation metrics used to assess the obtained results

	Settings
Model 1	stemming + uni-gram + TF-IDF
Model 2	light stemming + uni-gram + TF-IDF
Model 3	stemming + bi-gram + TF-IDF
Model 4	light stemming + bi-gram + TF-IDF
Model 5	stemming + uni-gram + BTO
Model 6	light stemming + uni-gram + BTO
Model 7	stemming + bi-gram + BTO
Model 8	light stemming + bi-gram + BTO

Table 2. Representation models

4.2. *Evaluation metrics*

To evaluate the performance of sentiment classification, we use standard classification performance metrics [23]: accuracy, precision, recall and F-measure. Table 3 shows a confusion matrix that describes how the three measures are defined and computed.

	Predicted positive	Predicted negative
Assigned Positive	A	b
Assigned Negative	C	d

Table 3. confusion matrix

- "a" represents the number of positive tweets correctly classified by the system to belong to the positive class.

- "b" represents the number of negative tweets classified by the system to belong to the positive class.

- "c" represents the number of positive tweets classified by the system to belong to the negative class.

- "d" represents the number of negative tweets correctly classified by the system to belong to the negative class.

Accuracy, recall and precision are calculated as follows:

$$\text{Accuracy} = (a+d) / (a+b+c+d)$$

$$\text{Recall} = a / (a+c)$$

$$\text{Precision} = a / (a+b)$$

For all the experiments, we applied an evaluation method based on 10-folds cross validation [24]. It consists of dividing the dataset into 10 equal-sized folds. Then, 10 iterations of training and testing was done using each time nine folds for training and the last fold for testing. The overall performance is measured by calculating the average of the 10 iterations.

4.3. Results

Table 3 presents obtained accuracy, recall and precision of the three classifiers using the different settings. The best accuracy was reached using SVM classifier with light stemming, bi-gram features and BTO weighing (78.31 %). K-NN was in the second rank with up to 77,20 % using stemming, bi-gram and TF-IDF weighing. NB comes in the third place with up to 76.20%.

Comparing the performance of the three classifiers, we can note that the best accuracy is achieved using SVM for almost all settings. NB classifier exceeds SVM in the case of combining stemming, bi-gram features and BTO weighing. Likewise, K-NN outperforms SVM when TF-IDF weighing is applied with stemming and bi-gram features.

Concerning the impact of the representation model on classification, we observed a substantial increase in the accuracy when root stemming is replaced by light stemming. This can be explained by the fact that reducing words to their three-letter root affects the semantics and even the sentiment orientation. Several words with different meanings might have the same root such as “ابتاع” which means “to buy” and its anonym “باع” which means “to sell”.

In addition, it was found that using bi-grams enhances the performance for all classifiers. This finding can be interpreted by the fact that bi-grams allow to handle negation. Actually, the bi-gram attaches the negation particle (such as “no/ لا” and “not/ ليس”) to the word which precedes it or follows it. This preserves the polarity phrases. For example, the expression “لا أحب” (I don’t like) is considered as one feature in the bi-gram model, but if we consider unigrams we will have as feature the word «أحب» (I like) and so the polarity is changed.

We have also examined the contribution of weighting schemes. As we can see on Table 4, SVM and NB classifiers achieved best results using BTO weighting. However, this weighting gives very poor results for the K-NN classifier on all representation models. In contrast, TFIDF weighting enables all classifiers to obtain good results.

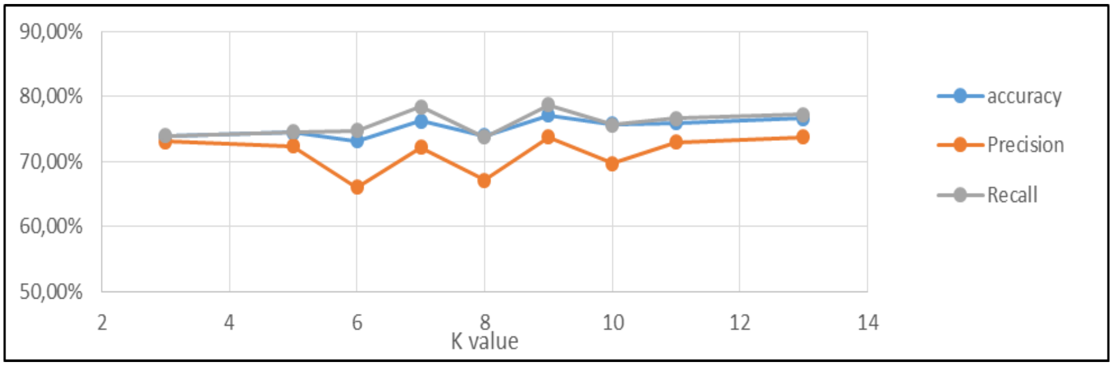

Fig. 1. Accuracy, Precision and Recall of K-NN

	SVM			NB			K-NN		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
stemming + uni-gram + TF-IDF	69,86%	86,80%	**76,50%**	67,08%	69,70%	70,08%	70,28%	77,27%	74,29%
light stemming + uni-gram + TF-IDF	71,58%	86,15%	**77,70%**	70,11%	70,56%	72,39%	73,18%	73,81%	75,28%
stemming + bi-gram + TF-IDF	69,97%	88,74%	77,11%	72,75%	71,65%	74,39%	73,83%	78,79%	**77,20%**
light stemming + bi-gram + TF-IDF	70,59%	90,91%	**78,21%**	72,77%	75,76%	75,59%	73,81%	73,81%	75,69%
stemming + uni-gram + BTO	63,80%	93,07%	**72,29%**	66,54%	74,89%	70,88%	46,86%	100%	47,39%
light stemming + uni-gram + BTO	68,85%	90,91%	**76,70%**	70,77%	75,97%	74,30%	46,57%	100%	46,79%
stemming + bi-gram + BTO	66,41%	94,95%	75,30%	73,58%	75,97%	**76,20%**	46,39%	100%	46,83%
light stemming + bi-gram + BTO	69,97%	93,29%	**78,31%**	73,39%	78,79%	76,90%	46,39%	100%	46,38%

Table 4. Results

Despite the positive results that have been achieved, the present work encountered a number of difficulties. The first problem that was faced is that most Twitter users do not care about spelling or grammatical errors when writing tweets. Therefore, the automatic correction of spelling mistakes and noise removal are important tasks that will certainly improve accuracy. The second difficulty is that communication in Twitter is usually carried out using dialectical Arabic rather than the more formal Modern Standard Arabic (MSA). The third challenge is that social media users use slang words. Moreover, they coin new words and expressions that do not belong to any dialect or language (e.g. “LOL” for “Laugh out loud”). The fourth problem that we encountered is that it is often difficult to identify irony and sarcasm. Sarcastic texts apparently have positive sentiments, but they are discreetly very ironic and negative.

Example:

”شكرا للفريق الوطني! أبهرتنا بإنجازاتك ككل عام هههه“

“Thank you to our National Team! You continue to fascinate us as every year. hehehh”

Finally, given that, the majority of Arabic proper nouns are derived from adjectives or can be used as adjectives, and taking into account the absence of capitalization in Arabic, it is often confusing in sentiment analysis to decide whether a given word is an adjective or the name of a person.

5. Conclusion and Future works

In this paper, our objective was to investigate sentiment classification in the Arabic language. In this context, we carried out a study on a 996 Arabic tweets collected automatically and annotated manually in two classes: positive and negative. We examined the effectiveness of some settings, namely stemming type, term weighting, and n-gram words. For classification purposes, we chose three common classifiers known for their efficiency: Naïve Bayes, Support Vector Machines and k-Nearest Neighbor classifiers. A comparison of the behavior of these three classifiers on our dataset shows that SVM was competitively effective (up to 78,31%). Moreover, the obtained results showed that the best setting for almost all classifiers on all data sets was the combination of light-stemming, bi-grams, and presence-based weighting. In our future works, we intend to incorporate more features that can improve classification performance such as n-gram character, Parts Of Speech, and lexical features.

References

1. B. Liu, ‘Sentiment analysis and opinion mining’, Synth. Lect. Hum. Lang. Technol., vol. 5, no. 1, pp. 1–167, 2012.
2. A. Mountassir, H. Benbrahim, and I. Berrada, ‘An empirical study to address the problem of unbalanced data sets in sentiment classification’, presented at the Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on, 2012, pp. 3298–3303.
3. M. Rushdi-Saleh, M. T. Martín-Valdivia, L. A. Ureña-López, and J. M. Perea-Ortega, ‘OCA: Opinion corpus for Arabic’, J. Am. Soc. Inf. Sci. Technol., vol. 62, no. 10, pp. 2045–2054, 2011.
4. M. A. Aly and A. F. Atiya, ‘LABR: A Large Scale Arabic Book Reviews Dataset.’, presented at the ACL (2), 2013, pp. 494–498.
5. S. R. El-Beltagy and A. Ali, ‘Open issues in the sentiment analysis of Arabic social media: A case study’, presented at the Innovations in information technology (iit), 2013 9th international conference on, 2013, pp. 215–220.
6. N. El-Makky et al., ‘Sentiment analysis of colloquial Arabic tweets’, 2015.
7. S. Kiritchenko, S. M. Mohammad, and M. Salameh, ‘SemEval-2016 Task 7: Determining sentiment intensity of english and arabic phrases’, presented at the Proceedings of the International Workshop on Semantic Evaluation (SemEval), San Diego, California, June, 2016.
8. S. R. El-Beltagy, ‘NileTMRG at SemEval-2016 Task 7: Deriving Prior Polarities for Arabic Sentiment Terms’, Proc. SemEval, pp. 486–490, 2016.
9. E. Refaee and V. Rieser, ‘iLab-Edinburgh at SemEval-2016 Task 7: A hybrid approach for determining sentiment intensity of Arabic Twitter phrases’, Proc. SemEval, pp. 474–480, 2016.
10. A. Htait, S. Fournier, and P. Bellot, ‘LSIS at SemEval-2016 Task 7: Using web search engines for English and Arabic unsupervised sentiment intensity prediction’, Proc. SemEval, pp. 469–473, 2016.
11. S. R. El-Beltagy, T. Khalil, A. Halaby, and M. Hammad, ‘Combining Lexical Features and a Supervised Learning Approach for Arabic Sentiment Analysis’, 2016.
12. J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, ‘Tackling the poor assumptions of naive bayes text classifiers’, presented at the ICML, 2003, vol. 3, pp. 616–623.
13. P. S. Dodds et al., ‘Human language reveals a universal positivity bias’, Proc. Natl. Acad. Sci., vol. 112, no. 8, pp. 2389–2394, 2015.
14. M. Abdul-Mageed and M. T. Diab, ‘SANA: A Large Scale Multi-Genre, Multi-Dialect Lexicon for Arabic Subjectivity and Sentiment Analysis.’, presented at the LREC, 2014, pp. 1162–1169.
15. E. Kouloumpis, T. Wilson, and J. D. Moore, ‘Twitter sentiment analysis: The good the bad and the omg!’, Icwsm, vol. 11, pp. 538–541, 2011.
16. E. Refaee and V. Rieser, ‘An Arabic Twitter Corpus for Subjectivity and Sentiment Analysis.’, presented at the LREC, 2014, pp. 2268–2273.
17. C. E. Shannon, ‘A mathematical theory of communication’, ACM SIGMOBILE Mob. Comput. Commun. Rev., vol. 5, no. 1, pp. 3–55, 2001.
18. H. Yu and V. Hatzivassiloglou, ‘Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences’, presented at the Proceedings of the 2003 conference on Empirical methods in natural language processing, 2003, pp. 129–136.

19. B. Pang, L. Lee, and S. Vaithyanathan, ‘Thumbs up?: sentiment classification using machine learning techniques’, presented at the Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, 2002, pp. 79–86.
20. A. Go, R. Bhayani, and L. Huang, ‘Twitter sentiment classification using distant supervision’, CS224N Proj. Rep. Stanf., vol. 1, p. 12, 2009.
21. T. Joachims, ‘Text categorization with support vector machines: Learning with many relevant features’, presented at the European conference on machine learning, 1998, pp. 137–142.
22. R. Tokuhisa, K. Inui, and Y. Matsumoto, ‘Emotion classification using massive examples extracted from the web’, presented at the Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, 2008, pp. 881–888.
23. F. Sebastiani, ‘Machine learning in automated text categorization’, ACM Comput. Surv. CSUR, vol. 34, no. 1, pp. 1–47, 2002.
24. T. M. Mitchell, ‘Machine learning’, McGraw Hill, 1997.