Egyptian Informatics Journal 20 (2019) 163–171

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

HILATSA: A hybrid Incremental learning approach for Arabic tweets sentiment analysis

Kariman Elshakankery *, Mona F. Ahmed

Faculty of Engineering, Cairo University, Egypt

ARTICLE INFO

Article history:
Received 19 October 2018
Revised 27 February 2019
Accepted 28 March 2019
Available online 4 April 2019

Keywords:
Sentiment analysis
Opinion mining
Hybrid approach
Arabic Tweets Sentiment Analysis
Sentiment classification
Self-learning

ABSTRACT

A huge amount of data is generated since the evolution in technology and the tremendous growth of social networks. In spite of the availability of data, there is a lack of tools and resources for analysis. Though Arabic is a popular language, there are too few dialectal Arabic analysis tools. This is because of the many challenges in Arabic due to its morphological complexity and dynamic nature. Sentiment Analysis (SA) is used by different organizations for many reasons as developing product quality, adjusting market strategy and improving customer services. This paper introduces a semi- automatic learning system for sentiment analysis that is capable of updating the lexicon in order to be up to date with language changes. HILATSA is a hybrid approach which combines both lexicon based and machine learning approaches in order to identify the tweets sentiments polarities. The proposed approach has been tested using different datasets. It achieved an accuracy of 73.67% for 3-class classification problem and 83.73% for 2-class classification problem. The semi-automatic learning component proved to be effective as it improved the accuracy by 17.55%.

1. Introduction

According to Liu [1], Sentiment Analysis (SA) and Opinion Mining field is concerned with analysing people's sentiments, opinions, emotions, attitudes and evaluations from document. Automatic SA identifies the writer emotion without the need to use the old manual ways. It is used by organizations and governorates to get feedback in order to improve their products and services. It is also used to predict the stock markets movements and elections results in real time based on events. SA is done on three levels. The first is the document level which assigns one class to the whole document. The second is the sentence level which assigns each sentence to a class. The third is the feature or aspect level which identifies the different aspects of the entity and computes a separate polarity for each one.

Lexicon based approach, Machine learning based approach and Hybrid approach are the three main used approaches for sentiment analysis. Lexicon based approach calculates the polarity of the document from the polarity of its words using dictionaries. It tends to have a good precision, but low recall in addition to the labour work needed to build the lexicons. This approach is adopted in [2]. Machine learning based approach depends on building classifiers as Support Vector Machine (SVM), Neural Network (NN), etc. The classifiers are trained to determine the document polarity as in [3] and [4]. However, the trained models are domain dependent and require a huge amount of training datasets. On the other hand the Hybrid approach combines both lexicon based and machine learning based approaches such that the trained model considers the lexicon based result in its features as in [5].

Being a valuable source of information for organizations, the need to make accurate SA is an important issue. Most opinions gathered from Arab social media is in dialectal Arabic, as the use of Modern Standard Arabic (MSA) in social media is rare. Research work that handles dialectal Arabic is still in its infancy and the accuracy still needs tuning. Also dialectal Arabic is a dynamic language where new words and terms are continuously being introduced by new generations, besides words usages changes over time. The proposed approach is based on the hybrid method that

* Corresponding author.
E-mail addresses: Kariman_Elshakankery@hotmail.com (K. Elshakankery), mona_farouk@eng.cu.edu.eg (M.F. Ahmed).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2019.03.002
1110-8665/

builds lexicons and uses them to train a machine learning based algorithm. It introduces a semi-automatic learning system that copes with the dynamic nature of the dialectal Arabic as it crawls twitter for new tweets and extends the lexicon by extracting words that do not exist in it and try to guess their polarities. Hence, the name HILATSA (Hybrid Incremental Learning approach for Arabic Tweets Sentiment Analysis). The main contribution of this work is introducing a sentiment analysis tool for Arabic tweets along with its ability to cope with the rapid change of words and their usages. As a part of the proposed approach some essential lexicons are built (words lexicon, idioms lexicon, emoticon lexicon and special intensified words lexicon). Besides, two lists including the intensification tools and the negation tools are created. All these will be available for public use to overcome the problem of lack of dialectal Arabic resources. The paper also investigated the effectiveness of using Levenshtein distance algorithm in SA to cope with different word forms and misspelling.

The rest of the paper is organized as follow. Section two describes briefly the related work done in the field of SA. Section three provides the details of the methodology along with the used datasets, lexicons and classifiers. Section four presents the results achieved on the different datasets and discusses them. Finally, section five provides the conclusion of the whole work.

2. Related work

Building a sentiment analysis tool for social media is an emergent task due to its wide spread usage and the valuable information that can be produced from it. While a lot of work was done on English, the research on Arabic is still in its early stages and many of the resources are not available for public use. Although some of the built lexicons could be extended, to the best of our knowledge none of them were proved to be able to cope with the dynamic nature of the language over time without the need of retraining. This section provides a quick overview of some of both Arabic and English tools.

Ibrahim et al. [5] built a system for MSA and Colloquial Arabic. They created two lexicons one for words and another one for idioms. They detected negation tools, intensifiers, wishes and questions. A Support Vector Machine (SVM) classifier was used to detect the sentence polarity.

Duwairi et al. [6] converted the emotions as fx1, fx2 to their corresponding words. Also, they converted both dialect sentences and Franco Arab words to their corresponding MSA. Their highest accuracy was achieved by using Naive Bayes (NB) classifier.

El-Makky et al. [7] used information gain for feature selection. The classification was done in two steps. First, determine whether subjective or objective then, classify the subjective ones further as positive or negative.

Al-Ayyoub et al. [8] built two different hierarchical classifiers with different core classifiers (SVM, Decision Tree (DT), K-Nearest Neighbour (KNN) and NB) in order to score the document.

Aldayel et al. [9] built a hybrid system for tweets in Saudi dialect. They used the output of the lexicon classifier and term frequency inverse document frequency (TF-IDF) to build their feature vector and train an SVM classifier. The hybrid approach improved the accuracy by 16%.

El-Masari et al. [10] created a web based tool with variable parameters. First, user selects a topic, then, a time interval to collect tweets regarding that topic. Finally, the system provides polarity distribution for the topic.

Abuelenin et al. [11] used the Information Science Research Institute Arabic stemmer (ISRI) and the cosine similarity algorithm in order to find the most similar word in lexicon and create a hybrid system to increase the accuracy of Egyptian Arabic.

Zhang et al. [12] proposed a new entity-level sentiment analysis method for Twitter with no manual labelling. First of all a lexicon was created manually in order to determine the tweet sentiment polarity. Then, opinion indicators as words and tokens were extracted using Chi Square based on the lexicon. Finally, a classifier was trained to identify the tweet sentiment polarity.

Nodarakis et al. [13] proposed a distributed parallel algorithm in Spark platform. In order to avoid the intensive manual annotation, tweets were labelled based on the emotions and the hashtags. A Bloom filter was applied to enhance the performance after building the feature vector. Finally, All k Nearest Neighbour (AkNN) queries are used to classify the tweets.

Guzman et al. [14] proposed a fine grained sentiment analysis system for application reviews. The system helps developers getting feedback regards their applications. First of all, the system extracts the application's features from the reviews. Then, it gives a general score to each feature based on the users' sentiments about the feature in the reviews.

Poria et al. [15] created a seven layer deep convolutional network for the aspect extraction sub-task. The system tags each word in the document as aspect or not aspect. Their approach gives better accuracy than both linguistic patterns and Conditional Random Fields (CRF).

Sabri et al. [16] empirically evaluated the use of Information Gain, Gini Index and Chi Square as Arabic features selection tools in addition to N-gram and Association Rules (AR) to build the feature vector to be used with Meta Classifier for Arabic SA tool. They concluded that Chi Square gave the best results while Meta Classifier combination of both N-gram and AR had the best performance.

This paper's main focus is to build a tool that is capable of evolving over time with less human interference. It builds the lexicons dynamically, and then expands them incrementally over time. The following section describes the proposed methodology in detail.

3. Methodology

HILATSA has three main parts: the lexicons, the classifier and the words learner. First of all a words lexicon was dynamically built from different datasets based on manual annotation. Each word in the lexicon has three values; Pos_Count, Neg_Count and Neu_Count, where Pos_Count is the number of times the word was considered positive, Neg_Count is the number of times the word was considered negative and Neu_Count is the number of times the word was considered neutral. An emotion lexicon was built from the popular emotions with their sentiments polarities. Also, a third lexicon was built for the most popular idioms. It was collected from websites and manually annotated as positive, negative and neutral. After that, a classifier was trained and tested. Finally, the semi-automatic learning part used the trained classifier to try to add new words to the lexicon and update old words weights. Fig. 1 shows HILATSA's block diagram.

3.1. Datasets

Five different datasets are used in order to build the lexicons, train the classifier, test and verify the system. A sixth dataset is used only for simulating and evaluating the lexicon expansion methodology. The datasets are:

1. ASTD [17]: It has around 10 K Arabic tweets from different dialects. Tweets are annotated as positive, negative, neutral and mixed.
2. Mini Arabic Sentiment Tweets Dataset (MASTD): More than 6 K tweets are collected using Twitter API, but only 1,850 tweets

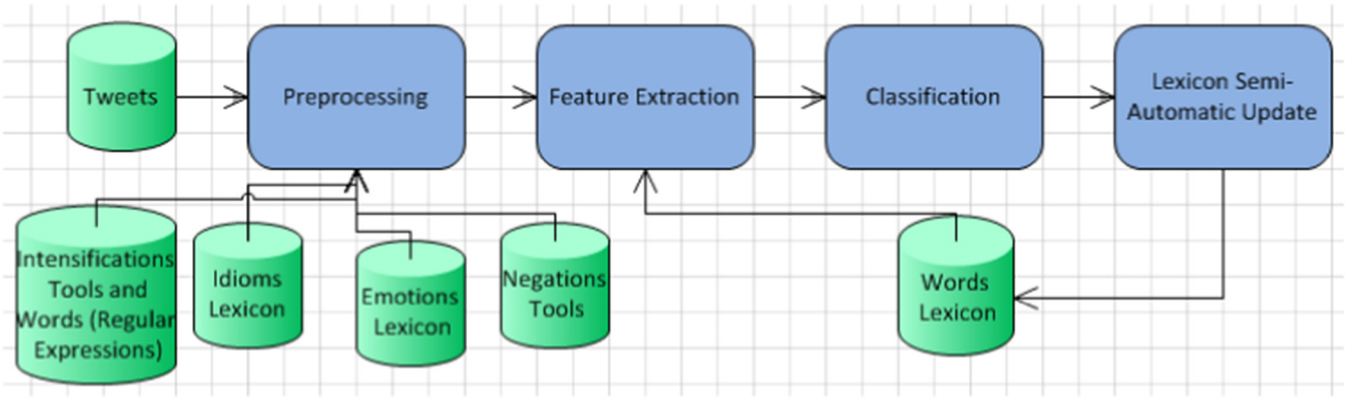

Fig. 1. HILATSA's Block Diagram.

are used. Others are neglected for duplications, sarcasm or conflict. Tweets are manually annotated to three classes neutral, negative and positive. Table 1 shows the dataset statistics. Fig. 2 shows tweets length versus number of tweets. Fig. 3 shows tweets n-gram distribution.

3. ArSAS [18]: It has more than 21 K tweets. The tweets are annotated as positive, negative, neutral and mixed.
4. Arabic Gold Standard Twitter Data for Sentiment Analysis [19] (GS): It has more than 8 K tweets. The tweets are annotated as positive, negative, neutral and mixed. The main problem with this dataset is that it only provides the tweets' ids along with their sentiments polarities, so the acquired tweets differ from time to time. We were able to acquire 4,191 tweets only out of the 8 K.
5. Syrian Tweets Corpus [20]: It has 2 K tweets in the Levantine dialect. Tweets are annotated as positive, negative and neutral.
6. Twitter dataset for Arabic Sentiment Analysis (ArTwitter) [21]: It has 2 K tweets written in both MSA and Jordanian dialect. Tweets are annotated as positive and negative.

For all datasets only the positive, negative and neutral tweets are used while the mixed ones were removed. Table 2 shows the datasets classes' distributions and the total number of acquired tweets from each dataset.

Table 1
MASTD Dataset Statistics.

Number of Tweets	1850
Median tokens per tweet	9
Max tokens per tweet	32
Avg. tokens per tweet	11
Number of tokens	20,067
Number of vocabularies	4,181
Number of positive tweets	415
Number of negative tweets	777
Number of neutral tweets	658

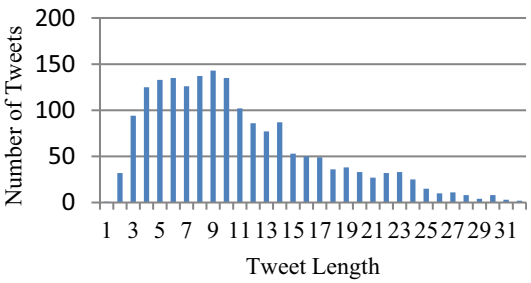

Fig. 2. Tweets Length.

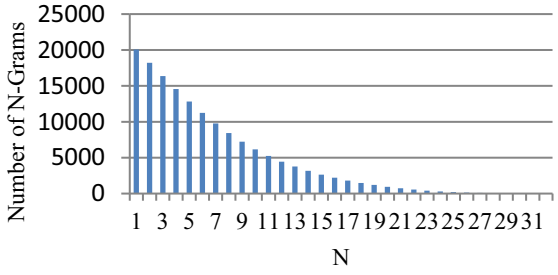

Fig. 3. Tweets n-gram.

Table 2
Datasets Distributions.

Dataset	Positive	Negative	Neutral	Total
ASTD	799	1,684	6,691	9,174
MASTD	415	777	658	1,850
ArSAS	4,643	7,840	7,279	19,762
GS	559	1,232	2,400	4,191
Syrian	448	1,350	202	2,000
ArTwitter	1,000	1,000	0	2,000

3.2. Building lexicons

In order to build the words lexicons (one for each fold), training datasets from the ASTD, MASTD, ArSAS, GS and the Syrian corpus are combined together in addition to EmoLex [22] as seed words. Then, the pre-processing steps (from next section) are applied in order to acquire stemmed unique words. Each word has three counters. First, the positive counter (Pos_Count) which represents the number of times the word appears in a positive tweet without negation or in a negative tweet with negation. Second, the negative counter (Neg_Count) which represents the number of times the word appears in a positive tweet with negation or in a negative tweet without negation. Finally, the neutral counter (Neu_Count) which represents the number of times the word appears in neutral tweets. Three scores (one per class) are calculated for each word. Only words with score higher than 0.75 for one of the three classes and length greater than 2 letters are added to lexicon where the word score for a certain class is calculated as the word frequency (counter) for this class divided by the total word frequency. The purpose of the score limitation is to ignore words of high frequency

Table 3
Examples from Words Lexicon.

Word	Pos_Count	Neg_Count	Neu_Count
ركن (Corner)	0	0	1
رفيع (Thin)	1	5	0
مساهمه (Contribution)	1	3	0

in all classes (i.e. stop words) while building the lexicon. Table 3 shows a small snap shot from the created lexicon.

An emotion lexicon is built from the popular emotions i.e.:D,:P, fx1, <3, fx2 with their polarities as positive or negative. Emotions were collected from [23] and [24] and then manually annotated. Table 4 shows examples from the emotions lexicon.

A small lexicon is created for special intensified words as لولوي, هه, هعهع, هاها along with their intensification forms expressed as regular expressions. These words are usually written by intensifying two letters repeatedly as هاهاهاها.

An idioms lexicon is built by collecting popular idioms and proverbs from websites. Then, they were manually annotated as positive, negative and neutral. Table 5 shows examples from the idiom lexicon.

3.3. Preprocessing

This phase is applied per tweet; it consists of a sequence of steps as follows:

1. Detecting and removing URLs.
2. Detecting and removing users' mentions.
3. Detecting whether the tweet has Hashtags or not.
4. Tokenization by splitting the tweets into tokens (i.e. words and emotions).
5. Detecting Emoji using the emotion lexicon to detect symbols as fx1, fx2,:D,:P,:O.
6. Removing non Arabic words and numbers.
7. Removing non-letters in words as كل3شهور
8. Normalization by removing diacritical marks, elongation and converting letters with different formats into a unique one as "أ" and "إ" replaced with "ي" , "ا" replaced with "ة" , "ى" replaced with "ئ" , "ه" and "ؤ" replaced with "لأ" ,"لآ" ,"ء" and "لإ" replaced with "پ" , "لا" replaced with "چ" , "ب" replaced with "ج" and "ڤ" replaced with "ف".
9. Detecting Intensification which can have variations of forms as duplicate consecutive characters (if more than two) or using intensification tools as, اوى جدا طحن or duplicate consecutive words or certain words as those collected in the special intensified words lexicon.
10. Detecting Negation either with negation tools as لا, مش, ليس, لن, لم or Egyptian negation patterns as in متكتبوش, ماتكتبوش, متكتبش, ماتكتبيش, ماتكتبوهاش and so on using an algorithm. The algorithm checks the word against different patterns as follows:
    - If a word starts with ما or م and ends with ش or وش or تش or كيش or هوش or ناش or همش or هاش or نيش or يش or كش or اش or توش or تكش, then the pattern will be removed from the word. After that it checks if the word (after removing the pattern) exists in the lexicon. If it does then, the word weight (as explained in section 3.4) is multiplied by the negation weight. If not and it ends with ت, then it replaces it with ه, and checks if it exists in the lexicon. If it does then, it multiplies the word weight by the negation weight. For example "مكتبتهاش" after removing the pattern it will be "كتبت". If it was not in lexicon then "ت" will be replaced with "ه" to be "كتبه" and it will be "كتب" in the next step by the stemmer.
    - If a word starts with لا then, it removes it (لا) and checks if the word (without this pattern) exists in lexicon. If it does then, it multiplies the word weight by the negation weight.
11. Stemming: A light stemmer is built based on the stemming algorithm in [25] to remove prefix, suffix from the word and to convert plural into singular. The objective of stemming is to downsize the lexicon in order to accelerate the search for the word to get its polarity.
12. Removing stop words by using the list provided by [26].
13. Detecting idioms using the idioms lexicon, the system searches for idioms in each tweet and merges their words together as a single token.
14. Detecting and removing people's names using the lexicon provided by Zayed et al. [27] to detect people's names as "ابراهيم، ريم" and remove them.

Table 4
Examples from Emotions Lexicon.

Emotion	Polarity
☺	1
☹	−1
U1F600	1

Table 5
Examples from Idioms Lexicon.

Idiom	Pos_Count	Neg_Count	Neu_Count
الفاضى يعمل قاضى (The one with free time, could be judgmental)	0	1	0
العين بصيره و اليد قصيره (The eye can see it, but the hand can't get it)	0	1	0
فى الحركة بركة (Motion is blessing)	1	0	0

3.4. Words weights

This step starts by searching for each word in the lexicon in order to compute its weights where for each word two weights are calculated: $w_{i_{neu-sub}}$ (the neutral-subjective weight for word w_i) as shown in Eq. (1) and $w_{i_{pos-neg}}$ (the positive–negative weight for word w_i) as in Eq. (2).

$$w_{i_{neu-sub}} = \frac{Neu\_Count_i - (Pos\_Count_i + Neg\_count_i)}{Neu\_Count_i + Pos\_Count_i + Neg\_count_i} \quad (1)$$

$$w_{i_{pos-neg}} = \frac{Pos\_Count_i - Neg\_count_i}{Pos\_Count_i + Neg\_count_i} \quad (2)$$

If a word was not found then the following steps are applied in order:

1. If a word starts with (ا- ه-س- تت), remove the first letter and search in lexicon i.e. "اكتب" after removing "ا" will be "كتب".
2. If it starts with (ن-ي-ت) then remove the first letter and search the lexicon. If not found, then try to replace the first letter with the other two letters and check the lexicon (try any of the alternative two letters) – تروح نروح– يروح.

If not found, Levenshtein Distance algorithm [28] is applied to get the similar word from the lexicon. The algorithm computes the distance between the word and each word in the lexicon. The distance is computed as the number of different, missing or added letters. The word in lexicon with the minimum distance is assumed

Table 6
Levenshtein Distance Algorithm Results.

Word	Matched Word in Lexicon	Distance	Number of Different Letters
مناسبه	مناسب	0.1667	1
انطلاق	انطلق	0.1667	1
اختبار	اختبر	0.1667	1
مانشستر	مانشيستر	0.143	1

to be the correct word given that the percentage of changes with respect to the original word length in the tweet is less than a certain threshold. Otherwise, the word is considered unknown and neglected. Table 6 represents some words with their matched words from the lexicon after applying the distance algorithm.

3.5. Building the feature vector

A feature vector is built, for each tweet, such that it does not depend on the words existence in the tweet, but on their weights (usages) and the tweet structure. It is built as follows:

1. The total neutral-subjective tweet sentiment: Each word weight is calculated as the percentage of the difference between its neutral counter and its subjective counter over its total appearance as illustrated in section 3.4. Then all words are summed together as shown in the following equation:

$$Total_{neu-sub} = \sum_{i=0}^{n} w_{i_{neu-sub}} \quad (3)$$

Where n is to the total number of words. The aim of the difference is that whenever a word has a similar counter in both classes its weight in the tweet score will be smaller.

2. The total positive–negative tweet sentiment. Each word weight is calculated as the percentage of the difference between its positive counter and negative counter over its total appearance as subjective as illustrated in section 3.4. Then all words are summed together as in the following equation:

$$Total_{pos-neg} = \sum_{i=0}^{n} w_{i_{pos-neg}} \quad (4)$$

3. The total neutral words weight.
4. The total positive words weight.
5. The total negative words weight.
6. A number to indicate whether the tweet is short, medium or long by dividing the number of its characters over the max number of characters allowed in a tweet (specified by Twitter API) i.e. 0 for short, 1 for medium and 2 for long. The tweet is considered short if the percentage of the number of characters in it over the max number of characters is less than 34%, medium if it is greater than or equal 34 but less than 67% and long otherwise.
7. Percentage of neutral words.
8. Percentage of subjective words.
9. Percentage of negative words.
10. Percentage of positive words.
11. Percentage of positive emotions
12. Percentage of negative emotions.
13. Whether the tweet includes emotions then this feature will have a value of 1 and 0 otherwise.
14. Whether the tweet includes idioms then this feature will have a value of 1 and 0 otherwise.
15. Whether the tweet includes URLs then this feature will have a value of 1 and 0 otherwise while the URLs themselves are removed as their contents are not useful.
16. Whether the tweet includes users' names then this feature will have a value of 1 and 0 otherwise while users' names are removed.
17. Whether the tweet includes Hashtags then this feature will have a value of 1 and 0 otherwise.

3.6. Classification

In this work SVM, L2 Logistic Regression and Recurrent Neural Network (RNN) classifiers are used. SVM is a linear classifier which creates a model to predict the class of the given data. It solves the problem by finding a general hyperplane with the maximum margin. It supports many formulas for classifications. This work uses C-SVM as the classifier where C is the penalty or the cost of wrong classification. The cost limits the training points of one class to fall in the other side (each side of the hyperplane represents a different class) of the plane. The higher the cost, the lower the number of points will be on the other side which in some cases may lead to over-fitting issue. Also, the lower the cost the higher will be the possibility of wrong classifications. When the cost function is high, the classifier selects a hyperplane (boundary) with a small margin which may give a good result regarding the training data, but may cause an overfitting issue and low accuracy regarding the testing data. On the other hand, when the cost function is low the classifier selects a hyperplane (boundary) with a wide margin which allows some training points from one class to fall in the other class area which can cause wrong classifications and low accuracy regarding the training data.

Logistic Regression is a discriminative classifier. It solves the problem by extracting weighted features, in order to produce the label which maximizes the probability of event occurrence. Regularization produces more general models and reduces overfitting by ignoring the less general features.

RNN is a neural network classifier which consists of layers each performs a specific function. It works on a sequence of input vectors and a sequence of output vectors instead of fixed number of inputs and outputs. It also keeps the history of the whole previous inputs. While basic NN uses the learnt computations (functions) to predict the output, RNN uses both the learnt computations along with a vector state to produce the output. This made the training as an optimization over the programs instead of functions according to [29].

All three classifiers use the feature vector created in the previous step to detect whether the tweet is positive, negative or neutral. While SVM was chosen for its good performance reported in the literature regards Arabic SA [30], Logistic Regression was chosen for its small training time and RNN was chosen for its promising results in sentiment analysis [31].

This work uses Chang et al. [32] SVM implementation and Fan et al. [33] L2 Regularization Logistic Regression implementation in addition to DL4J [34] RNN. Chang et al. [32] created LIBSVM which provides an implementation to the SVM classifier. It handles multi-class problems as a two-class problem by building k (k-1)/2 binary classifiers where k is the number of classes, and then it takes the majority vote for all those classes. In order to solve the optimization problem, it uses decomposition methods as Sequential Minimum Optimization (SMO). SMO is an algorithm for solving the quadratic programming problems by breaking a problem into its smallest sub-problems and solving them. Fan et al. [33] introduced LIBLINEAR library which is built based on LIBSVM but can handle large classification problems such as text mining. It supports L2 Regularization Logistic Regression. DL4J is an open source library that enables developers to build and configure their own neural networks in Java.

3.7. Lexicon Semi-Automatic update

The aim of this component is to add new words to the lexicon in order to keep it up-to-date with the language changes. It also updates old words weights to cope with the dynamic nature of the language and the changes on the word usage such as 'فظيع', this word was used to describe horrible things as in 'حادثه امبارح كانت فظيعه' which means 'Yesterday's accident was horrible'. However, today it is also used to describe something extraordinary as 'اللون ده فظيع و لايق اوى على المكان' which means 'This color is so good and perfectly goes with the place'. It is observed that dialectal Arabic, especially Egyptian dialect is a fast moving language so to be able to analyse it accurately, a dynamic system is recommended to keep continuously changing with the introduction of new terminologies and words usage changes. The words will be drawn from the new tweets. The system continuously gets new tweets, extracts the new words (words that does not exist in the lexicon) and tries to label them with a suitable sentiment polarity according to the whole tweet classification in addition to updating the weights for the old words (words that already exist in the lexicon) by increasing the word counter that corresponds to the new tweet's class. The following procedure is followed to get the new tweets labels and hence the labels of the new words in these tweets or the possible change of the already existing words weights. Given a tweet with unknown polarity, the system classifies it with 3-Class classifier. If the tweet is neutral then the system will check that it does not contain any emotions or idioms and that the difference between its neutral weight and subjective weight exceeds a certain threshold in order to consider it correct. Otherwise the system will add it to a list to be manually annotated before extracting the new words from it. If the tweet is subjective it will be classified for a second time with a 2-Class classifier. If both produce the same classification then the system will check that the difference between its positive weight greater than its negative weight in order to consider it correct if the classification result was positive and vice versa. Otherwise it will be added to a list to be manually annotated before extracting the new words from it. The objective of these restrictions is to decrease the probability of learning wrong words. Fig. 4 represents the lexicon semi-automatic update algorithm.

4. Experiments & results

This section describes the results of both the classification module and the Lexicon semi-automatic update module. The tool implementation is in Java using LIBSVM, LIBLINEAR and DL4J libraries.

4.1. Classification module

Using 10-fold cross validation with ASTD, MASTD, ArSAS, GS and the Syrian Corpus training datasets the word lexicon is built (different lexicon for each fold excluding the testing part) and the classifiers are trained. This section describes the results in terms of accuracy and average F1 score (Avg. F1). In addition, comparisons of the achieved results with other available work are also provided.

Table 7 shows HILATSA's results for the unbalanced datasets in a 3-Class classification problem. The SVM classifier accuracy tends to be higher than both L2-Logistic Regression (L2R2) and RNN. On the other hand, L2R2 classifier tends to have a better F1 score.

Table 8 shows HILATSA's results for the balanced datasets in a 3-Class classification problem. The SVM classifier tends to have higher accuracy and F1 score than both L2R2 and RNN.

Table 9 shows the results of the unbalanced datasets in a 2-Class classification problem. Considering HILATSA, L2R2 tends to have higher accuracy and better F1 score. On the other hand, although Al-Azani et al. [35] ensemble classifier has a better accuracy regarding the Syrian corpus, HILATSA has a better F1 score. Given that the classifier used by Al-Azani et al. [35] was trained

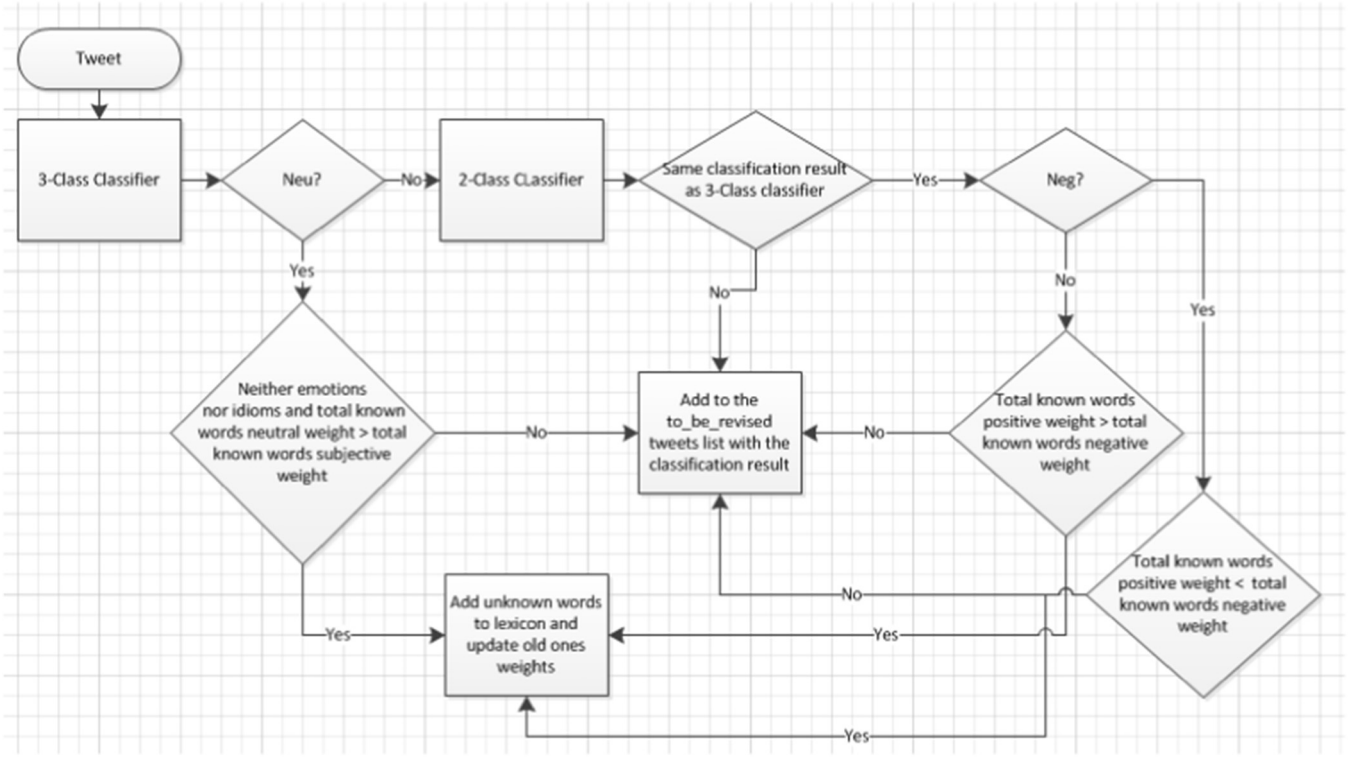

Fig. 4. Lexicon Semi-Automatic Update Algorithm.

Table 7
Unbalanced 3-Class Classification Problem.

	ASTD		MASTD		ArSAS		GS		Syrian Corpus	
	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1
L2R2	62.94	36.43	71.64	**68.57**	**66.70**	63.82	50.50	**37.29**	72.01	**48.33**
SVM	**73.11**	37.61	**71.69**	68.25	66.39	**63.85**	**54.64**	33.00	73.01	47.13
DL RNN	72.30	**40.25**	65.25	61.58	65.67	62.71	51.29	36.71	**73.67**	47.39

Table 8
Balanced 3-Class Results.

	ASTD		MASTD		ArSAS		GS		Syrian Corpus	
	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1
HILATSA (L2R2)	49.92	48.96	**66.83**	**66.55**	63.26	63.27	50.24	49.93	53.71	51.96
HILATSA (SVM)	**52.95**	52.03	63.74	63.54	**63.44**	**63.50**	**51.51**	**51.32**	52.83	51.97
HILATSA (RNN)	51.77	**52.21**	48.94	44.89	63.25	63.06	50.42	49.90	**56.17**	**56.01**

Table 9
Unbalanced 2-Class Results.

	ASTD		MASTD		ArSAS		GS		Syrian Corpus	
	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1
HILATSA (L2R2)	74.41	66.30	**83.73**	**82.03**	80.87	79.09	**68.09**	**58.29**	81.96	**74.50**
HILATSA (SVM)	74.98	**69.72**	72.54	69.39	**81.52**	**80.07**	67.13	57.33	81.28	74.41
HILATSA (RNN)	67.45	68.75	70.68	67.22	81.17	79.54	66.18	57.91	81.62	74.32
Al-Azani et al. [35]	–	–	–	–	–	–	–	–	**85.28**	63.95
Dahou et al. [36]	**79.07**	–	–	–	–	–	–	–	–	–

using Synthetic Minority Oversampling Technique (SMOT) and word embedding model using 190 million words, HILATSA with a comparatively limited lexicon is considered competitive.

Table 10 shows the results of the balanced datasets in a 2-Class classification problem. Both SVM and RNN tend to have better accuracy than L2R2. On the other hand, SVM tends to have better F1 score. In addition, for ASTD, SVM accuracy was so close to Dhou et al. [36]. This is considered a superlative performance for HILATSA given that the classifier used by Dhou et al. [36] was trained using word embedding model with more than 3 million words.

In summary, SVM tends to have better accuracy in most cases which is consistent with [30]. That is due to the kernel trick which allows a better accuracy with higher dimension data. In addition to that, it is less sensitive to outliers and less prone to overfitting. Also, 3-Class sentiment analysis is more challenging than 2-Class which is expected. In addition, in most cases F1 score is higher for balanced datasets than unbalanced datasets.

4.2. Lexicon Semi-Automatic update module

Using the dataset (ArTwitter) from [21] a classifier is trained using the same lexicons from the previous part (built from datasets other than ArTwitter). After that, the tweets are classified using 10-fold cross validation for training and testing. Then the words learner is used to learn words and update the lexicon. Finally, the tweets are classified again after updating the lexicon.

Table 11 shows the results for the testing dataset (ArTwitter) before and after the lexicon update (using 10-fold cross validation for both cases). The update was done with three different classifiers on three independent experiments in order to compare the effect of classifier type with respect to accuracy. The results show that HILATSA's accuracy and average F1 score increased after the learning phase. However, the increase was slightly higher while using SVM classifier in the learning phase for lexicon update than RNN. RNN achieved the highest accuracy before the learning phase. Also, its accuracy is still the highest after the update. The good

Table 10
Balanced 2-Class Results.

	ASTD		MASTD		ArSAS		GS		Syrian Corpus	
	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1
HILATSA (L2R2)	72.47	72.18	**78.17**	**78.07**	78.75	78.74	66.09	65.92	66.0	73.84
HILATSA (SVM)	75.19	**75.70**	69.02	68.65	**79.87**	**79.86**	66.82	66.57	68.25	**75.82**
HILATSA (RNN)	74.94	75.38	69.15	68.82	79.39	79.38	**66.91**	**66.74**	**75.5**	75.33
Dahou et al. [36]	**75.9**	–	–	–	–	–	–	–	–	–

Table 11
Lexicon Update Results.

	Before Update		After L2R2 Update		After SVM Update		After RNN Update	
	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1	Accuracy	Avg. F1
HILATSA (L2R2)	65.45	64.40	81.0	80.91	83.0	82.97	82.8	82.76
HILATSA (SVM)	67.65	67.35	78.95	78.41	81.5	81.37	81.25	81.12
HILATSA (RNN)	**68.45**	**68.32**	**83.15**	**82.98**	**85**	**84.95**	**84.6**	**84.55**

Table 12
A Sample of new words added to words lexicon.

Tweet	Word	HILATSA's Suggestion	Automatic	Actual Classification
الماء اساس الحياه (Water is the base of life)	اساس	Positive	Yes	Positive
اللهم ارزقنا مكارم الاخلاق (Oh Allah, grant us good manners)	مكارم	Positive	Yes	Positive
اللهم إني اعوذ بك من الذنوب التي تزيل نعمتك وتحول عافيتك وتحبس رزقك وتعجل بغضبك (Oh Allah, I seek refuge in you from the sins that remove your grace, transform your health, keep your livelihood and hasten your anger)	غضب	Negative	No	Positive
اللهم اهدنا صراطك المستقيم (Oh Allah, guide us to the right path)	مستقيم	Positive	Yes	Positive
انته جميل جدا ربنا يثبتك واتمني ان كل المسلمين تكون اخلاقياتهم في حياتهم العاديه قدوه ودعوه لكل غير المسلمين (You are beautiful, may Allah let you be the same and I wish that all Muslims may have morals in their daily life that non-Muslims look up to)	قدوه	Negative	No	Positive

performance of RNN with (ArTwitter) [21] is due to the NN ability to generalize so when it is used with a new dataset that wasn't used in building the lexicon it achieved the best results.

Table 12 shows examples of the new words the system is able to learn from the testing dataset (either automatically or manually) with their deduced sentiment polarity written in the column named "HILATSA's Suggestion". Thus, new words will incrementally be added to the lexicon. For example, in 'الماء اساس الحياه' which means 'Water is the base of life', HILATSA classified it as a positive tweet then it passed the validation phase so the new word 'اساس' (literary means base) which didn't exist in the lexicon is added with positive weight (Pos_Count = 1, Neg_Count = 0 and Neu_Count = 0). In addition to that, the old words' ('ماء' ('Water') and 'حياه' ('Life')) positive weights are increased. On the other hand,' انته جميل جدا ربنا يثبتك واتمني ان كل المسلمين تكون اخلاقياتهم في حياتهم العاديه قدوه ودعوه لكل غير المسلمين' which means 'You are beautiful, may Allah let you be the same and I wish that all Muslims may have morals in their daily life that non-Muslims look up to', HILATSA classified it as negative tweet, but it did not pass the validation phase so HILATSA added it to the to_be_revised tweets list, that includes all tweets that have to be manually revised, with suggested classification as negative tweet. The tweet was manually revised then used by HILATSA in order to update the lexicon. Table 13 shows example of tweets that passed the automatic-learning phase (don't need manual annotation) so that HILATSA considered them correct versus their correct classification. Table 14 shows examples of tweets that did not pass the automatic-learning phase and have to be manually annotated with their HILATSA's classification suggestion and the correct classifications.

Table 13
A sample of tweets that passed the automatic-learning phase with HILATSA's classification vs actual classifications.

Tweet	HILATSA's Classification	Actual Classification
هذا ليس مضحك ، بل مخزى ، يا للخزى و العار (This is not funny, it is disgrace. How shameful)	Negative	Negative
والله هبل في هبل (I swear it is all nonsense)	Negative	Negative
اللهم ارزقنا الصحبه الصالحة (Oh Allah, grant us good fellowship)	Positive	Positive
لانهم يمكن يكونو قاهرينك (As they may anger you)	Positive	Negative

Table 14
A sample of tweets that didn't pass the automatic-learning phase with their HILATSA's classification suggestions vs their actual classifications.

Tweet	HILATSA's Suggestion	Actual Classification
برنامج معوق ومشاركين معوقين:) (Bad show and bad participants)	Negative	Negative
عجبتنى الحلقه (I liked the episode)	Negative	Positive
الله يهدي الجميع ويوفقك (May Allah guide every one and grant you success)	Positive	Positive
مرررعبه (Terrifyyyying)	Negative	Negative

5. Conclusions

Although a lot of work was done for sentiment analysis, not enough work targeted Arabic. Besides the different dialects and the morphological complexity, Arabic has different grammar, structures and linguistic features other than English which complicates its analysis.

Because of the wide spread of social media, a lot of data is available. However, the public resources are still limited. Another issue with the resources is the change of word usage over time, word meaning differs and new words are added from time to time. Automatically updating the lexicon has a good impact on the system to learn new words. Still the learning process must have some restrictions or the system may learn wrongly. Manual annotating the tweets to be used by the system to extract and learn new words is still time consuming process, but proved to be useful when combined with the automatic part. This work presents a hybrid system for Arabic Twitter sentiment analysis that provides high accuracy and reliable performance in a dynamic environment where language changes is an everyday issue that requires language analysis systems to intelligently cope with these continuous changes.

References

[1] Liu B. Sentiment analysis and opinion mining. Synth Lect Hum Lang Technol 2012;5(1):1–167.
[2] Taboada M, Brooke J, Tofiloski M, Voll K, Stede M. Lexicon-based methods for sentiment analysis. Comput Linguist 2011;37(2):267–307.
[3] Aldahawi H, Mining and analysing social network in the oil business: twitter sentiment analysis and prediction approaches. PhD Thesis, Cardiff University, 2015.
[4] Jahren BE, Fredriksen V, Gambäck B, Bungum L. NTNUSentEval at SemEval-2016 Task 4: combining general classifiers for fast twitter sentiment analysis. Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), 2016. p. 103–8.
[5] Ibrahim HS, Abdou SM, Gheith M. Sentiment analysis for modern standard Arabic and colloquial. Int J Nat Lang Comput (IJNLC) 2015;4(2).
[6] Duwairi RM, Marji R, Sha'ban N, Rushaidat S. Sentiment analysis in Arabic tweets. In: 5th International conference on information and communication systems. ICICS; 2014.
[7] El-Makky N, Nagi K, El-Ebshihy A, Apady E, Hafez O, Mostafa A, et al. Sentiment analysis of colloquial arabic tweets. In: ASE BigData/SocialInformatics/PASSAT/BioMedCom 2014 Conference. Harvard University; 2014. p. 1–9.

[8] Al-Ayyoub M, Nuseir A, Kanaan G, Al-Shalabi R. Hierarchical classifiers for multi-way sentiment analysis of arabic reviews. Int J Adv Comput Sci Appl 2016;7(2).
[9] Aldayel HK, Azmi A. Arabic tweets sentiment analysis-a hybrid scheme. J Inf Sci 2015;42(6):782–97.
[10] El-Masri M, Altrabsheh N, Mansour H, Ramsay A. A web-based tool for Arabic sentiment analysis. Procedia Comput Sci 2017;117:38–45.
[11] Abuelenin S, Elmougy S, Naguib E. Twitter sentiment analysis for Arabic tweets. In: International conference on advanced intelligent systems and informatics. Cham: Springer; 2017. p. 467–76.
[12] Zhang L, Ghosh R, Dekhil M, Hsu M, Liu B. Combining lexicon-based and learning-based methods for twitter sentiment analysis. In: Technical Report HPL-2011. HP Laboratories; 2011. p. 89.
[13] Nodarakis N, Sioutas S, Tsakalidis AK, Tzimas G. Large scale sentiment analysis on twitter with spark. EDBT/ICDT workshops, 2016. p. 1–8.
[14] Guzman E, Maalej W. How do users like this feature? a fine grained sentiment analysis of app reviews. In: Requirements Engineering Conference (RE), 2014 IEEE 22nd International. IEEE; 2014. p. 153–62.
[15] Poria S, Cambria E, Gelbukh A. Aspect extraction for opinion mining with a deep convolutional neural network. Knowl-Based Syst 2016;108:42–9.
[16] Sabri B, Saad S. Arabic sentiment analysis with optimal combination of features selection and machine learning approaches. Res J Appl Sci, Eng Technol 2016;13:386–93. https://doi.org/10.19026/rjaset.13.2956.
[17] Nabil M, Aly M, Atiya A. ASTD: Arabic sentiment tweets dataset. In: Proceedings of the 2015 conference on empirical methods in natural language processing. p. 2515–9.
[18] Elmadany AA, Hamdy Mubarak WM. ArSAS: an Arabic speech-act and sentiment corpus of tweets. OSACT 3: the 3rd workshop on open-source arabic corpora and processing tools, 2018. p. 20.
[19] Refaee E, Rieser V. An Arabic twitter corpus for subjectivity and sentiment analysis. LREC, 2014. p. 2268–73.
[20] http://saifmohammad.com/WebPages/ArabicSA.html (Last accessed: Monday 18-02-2019 16:22).
[21] Abdulla N, Mahyoub N, Shehab M, Al-Ayyoub M. Arabic Sentiment Analysis: Corpus-based and Lexicon-based. IEEE conference on Applied Electrical Engineering and Computing Technologies (AEECT 2013), 2013.
[22] http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm.
[23] http://www.unicode.org/emoji/charts/full-emoji-list.html (Last accessed: Monday 18-02-2019 16:22).
[24] http://emojitracker.com/ (Last accessed: Monday 18-02-2019 16:22).
[25] Khader S, Sayed D, Hanafy A. Arabic light stemmer for better search accuracy. World Acad Sci, Eng Technol, Int J Soc, Behav, Educ, Econ, Bus and Ind Eng 2016;10(1):3577–85.
[26] Shoukry A, Rafea A. Arabic sentence level sentiment analysis. In: Proceedings of the 2012 international conference on collaboration technologies and systems. CTS; 2012. p. 546–50.
[27] Zayed O, El-Beltagy S. Named entity recognition of persons' names in Arabic tweets. Proceedings of the international conference recent advances in natural language processing, 2015. p. 731–8.
[28] Levenshtein V. Binary codes capable of correcting deletions, insertions and reversals. Soviet Phys Doklady 1966;10:707–10.
[29] http://karpathy.github.io/2015/05/21/rnn-effectiveness/ (Last accessed: Monday 18-02-2019 16:22).
[30] Kaseb Gehad S, Ahmed Mona F. Arabic sentiment analysis approaches: an analytical survey. Int J Sci Eng Res 2016;7(10):1871–4.
[31] Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng A, Potts C. Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of the 2013 conference on empirical methods in natural language processing, 2013. p. 1631–42.
[32] Chang C, Lin C. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2013;2:1–39.
[33] Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: a library for large linear classification. J Mach Learn Res 2008;9:1871–4.
[34] https://deeplearning4j.org/ (Last accessed: Monday 18-02-2019 16:22).
[35] Al-Azani S, El-Alfy ESM. Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short Arabic text. Procedia Comput Sci 2017;109:359–66.
[36] Dahou A, Xiong S, Zhou J, Haddoud MH, Duan P. Word embeddings and convolutional neural network for Arabic sentiment classification. Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers, 2016. p. 2418–27.