Journal of Engineering and Applied Sciences 13 (17): 7329-7340, 2018
ISSN: 1816-949X

A Large-Scale Arabic Sentiment Corpus Construction Using Online News Media

[1]Ahmed Nasser and [2]Hayri Sever
[1]Department of Control and Systems Engineering, University of Technology, Baghdad, Iraq
[2]Department of Computer Engineering, Hacettepe University, Ankara, Turkey

Abstract: Within computer-based technologies, the usage of collected data and its size are continuously on a rise. This continuously growing big data processing and computational requirements introduce new challenges, especially for Natural Language Processing (NLP) applications. One of these challenges is maintaining massive information-rich linguistic resources which are fit with the requirements of the big data handling, processing and analysis for NLP applications such as large-scale text corpus. In this research we present a large-scale sentiment corpus for the Arabic language called GLASC which is built using online news articles and metadata shared by the big data resource GDELT. Our GLASC corpus consists of a total number of 620,082 news article which are organized in categories (Positive, negative and neutral). Besides that, each news article within our corpus has a sentiment rating score in the range between-1 and 1. We have also carried out some experiments on our corpus, using machine learning algorithms to generate a sentiment classifier for document-level Arabic sentiment analyses. For training the sentiment classifier we generated different datasets from our corpus using different feature extraction and feature weighting method. We performed a comparative study, involving testing a wide range of classifiers that commonly used for sentiment analysis task and in addition we investigated several types of ensemble learning methods to verify its effect on improving the classification performance of sentiment analysis by using different comprehensive empirical experiments.

Key words: Sentiment analysis, large-scale corpus, bigdata, machine learning, ensemble learning, experiments

INTRODUCTION

The process of examining and identifying the sentiment or emotions that realis behind the words is called Sentiment Analysis (SA). The main purpose of SA is to capture the tone of feeling that expressed by the words used within the text. The terms of sentiment analysis (Nasukawa and Yi, 2003) and idea mining (Dave *et al*., 2003) first appeared in 2003. Elliot (1992) and Ortony and Turner (1990) have carried out the primitive SA method which depends on the existence of the effective word. Although, SA consists of hybrid studies by means of combining the methods; it mainly consists of two methods: these methods are Machine Learning (ML) based methods (Pang and Lee, 2008) and Lexicon Based (LB) methods (Taboada *et al*., 2011). SA or often called Opinion Mining (OM) utilizes different methods for information extraction such as text analysis, Natural Language Processing (NLP) and computational linguistics (Pang *et al*., 2002). SA or Opinion Mining (OM) is used in wide range of area such as; evaluation, social media marketing and customer service. In general, SA aims to identify the attitude of the speaker/writer or sentiment polarity of textual contents for a particular title or subject.

Nowadays, the massive and the rapid growth of the big data internet resources handling introduced a new set of difficulties, especially in articial intelligence applications such as NLP (Ptaszynski *et al*., 2014). One of the important difficulties in such applications is maintaining large information-rich resources such as a large-scale text corpus which is considered as the most vital linguistic resources that can be used for training and evaluation many NLP ML applications such as SA (Ptaszynski *et al*., 2014).

In NLP applications large-scale resources become an essential demand for ensuring the performance and the robustness of these applications (Gandomi and Haider, 2015).

The importance of corpus size with regard to the number of word in the corpus is investigated by Baayen (2001) where the authors noticed that the within a given corpus the appearance probability of a particular words follows the distribution that achieved with Zip's

Corresponding Author: Ahmed Nasser, Department of Control and Systems Engineering, University of Technology, Baghdad, Iraq

law (Zipf, 1935) which state that "Within a corpus the words occurrences frequencies tend to decrease in a quadratic-like manner".

If we generated a list consist of the all unique words within a certain corpus together with its corresponding occurrence frequencies and then sort this list based on the occurrence frequencies of the words descendingly. We can see that the last word in the list tends to appear two times lesser than the previous word in the list and so on. This can prove the relation between the corpus size and the number of words within the corpus. So that in the case of corpus size is small, the probability of many words to be not appeared in this corpus is high and vice versa.

To address the lack of availability of such large-scale resource for the Arabic language, we introduce our large-scale Arabic sentiment analysis corpus that is built using online news media and utilizing the metadata that produces by the big data resource GDELT. Our corpus consists of a total 620,082 news article. All these news articles within the corpus have sentiment score between 1-1 and organized in form of three categories (positive, negative and neutral).

We can summarize the contributions in this research as follows. First, generating the largest up-to-date resource for the Arabic language which we believe it can help to improve not only sentiment analysis application but also a wide spectrum of NLP applications for the Arabic language in general.

Second, our large-scale sentiment corpus is used to generate four datasets by using different feature extraction and feature weighting methods. These datasets can be used to build and evaluate ML-based sentiment analysis systems.

Third, we have carried out some experiments on the datasets generated from our large-scale corpus, using ML algorithms to train the sentiment classifier. We have forced on using the ML classification methods that are widely used in sentiment analysis works on the literature such as Support Vector Machine (SVM), Hidden Markov Model (HMM), Neave Bayes (NB), Neural Network (NN) and K-Nearest Neighbors (KNN) to determine the sentiment polarity. Then we conduct a comparative assessment of the performance of these different classification methods by using the different datasets generated from our large-scale corpus.

The fourth contribution of this study is to verify the effectiveness of using ensemble learning for sentiment classification. Due to the increase of interest in using classifier ensemble techniques (which is used to combine multiple classification models in order to increase and enhance the classification accuracy) in the last years (Wang *et al*., 2014), We investigate the effectiveness of enhancing the classification accuracy in sentiment analysis applications using popular ensemble methods (Bagging, Boosting and Random Subspace and staking) based on five base learners (SVM, HMM, NB, NN and KNN) for sentiment classification.

Literature review: The popular data resource that commonly used for Arabic sentiment analysis works in the literature can be summarized as following.

Saleh *et al.* (2011) presented OCA which is an Arabic opinion corpus consist of 500 text documents where each of the document represents a movie review. These movie reviews were collected from 15 different web pages and organize into two categories positive and negative movie reviews where each category contains 250 reviews.

Abdul-Mageed and Diab (2012) presented a multi-genre corpus called Awatif for Arabic sentiment analysis. This corpus manually annotated through crowdsourcing from three different resources (Penn Arabic Treebank (PATB), Wikipedia talk pages and seven different web forums) and contains 2855 reviews.

Aly and Atiya (2013) presented LABR which is a large-scale book review dataset for Arabic sentiment analysis. This dataset contains 63000 reviews collected from GoodReads Arabic book review website and each review has a rating from 1-5 which refers to negative and positive reviews respectively.

SAMAR is presented by Abdul-Mageed *et al.* (2014) which is a system for Arabic social media sentiment analysis based on morphological features. SAMAR used to evaluate three different multi-domain TAGREED (TGRD), TAHRIR (THR) and MONTADA (MONT) which are collected from Twitter, Wikipedia TalkPages and Arabic forums, respectively.

A human annotated dataset for Arabic book review called HAAD is presented by Al-Smadi *et al.* (2015) and used for aspect-based sentiment analysis. HAAD dataset contains a manually annotated 2389 Arabic book reviews with an aspect terms.

By El-Sahar and El-Beltagy (2015) a large multi-domain Arabic Sentiment Analysis dataset is generated from different reviewing websites. This dataset contains 33,000 reviews annotated from movies, hotels, restaurants and products reviews websites. Table 1 provides a comparison between the popular Arabic data resources used in the most sentiment analysis researches that available in the literature.

Table 1: A comparison between Arabic sentiment analysis data resources

Corpus/dataset	Citations	Size	Data source	Categories
OCA	118	500.000	Movie reviews	Positive negative
Awatif	80	2.855	Penn Arabic treebank,	Positive negative
			Wikipedia talk pages and web forums	neutral
LABR	39	63.000	goodreads	1-5 rating
SAMAR(TGRD)	139	3.015	Twitter	Positive negative
SAMAR(THR)	139	3.008	Wikipedia talk pages	Positive negative
SAMAR(MONT)	139	3.097	Arabic forums	Positive negative
HAAD	16	2.389	Book reviews	Positive negative
Multi-domain Arabic Sentiment	26	32.338	Movies, hotels, restaurants	conflict neutral
Analysis datasets			and products reviews	Positive negative
			GDELT, Arabic news	Positive negative
Our large-sale arabic sentiment	-	620.082		neutral
Corpus GLASC				Score-1-1

MATERIALS AND METHODS

Large-scale Arabic sentiment corpus generation: In this study, we reviewed the data resources that we utilized. And explained the method we used for generating our large-scale Arabic sentiment corpus. In additional we also carried out some experiments to verify the quality of our corpus.

GDELT: There are many incidents happening throughout the world in the last 24 h and that is worthy of being news in the mainstream media. These events which are captured and updated every 15 min from 1979 to present by GDELT "(Global Database of Events, Language and Tone)" project, can only be defined as a big data. GDELT put all these data at the disposal of all researchers worldwide as open-source big data (Sagi and Labeaga, 2016).

GDELT is scanning the world's mainstream news media as well as the social media, multimedia objects and the environment of digital library characteristics such as DTIC, JSTOR to obtain GDELT codified metadata. This metadata is then released as an open data stream, updated Every 15 min GDELT is scanning the world's mainstream news media as well as the social media, multimedia objects and the environment of digital library characteristics such as DTIC, JSTOR to obtain GDELT codified metadata. This annotated metadata stored and indexed in GDELT databases (Sagi and Labeaga, 2016).

If the language of the scanned source text is one of 65 different languages other than English, GDELT source language identifier is triggered. Currently for 50 languages out of 15 languages (Arabic, Basque, Catalan, Chinese, French, Galician, German, Hindi, Indonesian, Korean, Pashto, Portuguese, Russian, Spanish, Urdu), news text is depicted to English in real time and natural language processing mechanisms are run to record the inferred assets and the tags and metrics for each entity in the database. 15 languages are directly passed directly into the analysis process without translating the English language through existing dictionary sub-structures, thus allowing analysis without loss and incoherency due to translation (Anonymous, 2015).

In general it is seen that the requirements which form the basic pillars of the concept of big data and included in the literature as 5 V, (Volume, Velocity, Variety, Veracity, Value) are found in the GDELT. Also by being an open source of big data, GDELT will be used as a basic data source for the academic world for decision support processes in the near future, so that, the researchers, executive powers, conjuncture-based decision-making and investment specialists will be able to capture the moments in the world.

GDELT presents essentially two main datasets: "Events" and "Global Knowledge Graph (GKG)". These datasets use "Conflict and Mediation Event Observations" (CAMEO) coding for recording events and saved in CSV file format.

The GKG database keeps track of people organizations, companies, positional data and the data tagged with theme and sentiment tags, from each news source scanned. In our study, we used GKG dataset to obtain URLs of news and their tone values. The tone value between +100 and -100 that represents the sentiment score related to a specific news article.

To interact with the databases and datasets that offered by GDELT, Google big query is used together with Structure Query Language (SQL). The data obtained from GDELT databases the can be accessed via. Google's cloud storage and download in form of CSV files (Sagi and Labeaga, 2016).

The process of generating a large-scale sentiment corpus using GDELT: The process of generating our GDELT Large-scale Arabic Sentiment Corpus (GLASC) is illustrated in Fig. 1. This task of corpus generating is done as follows:

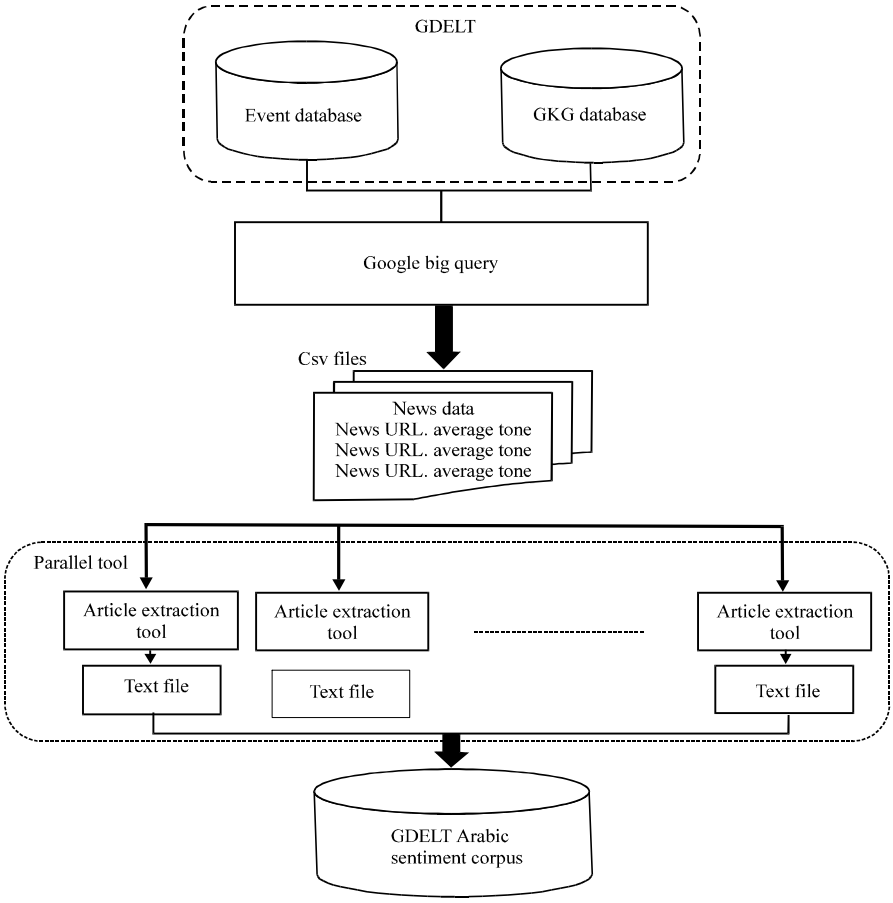

Fig. 1: Our GLASC corpus generation process

Firstly, we used SQL query to fetch the data that related to Arabic news from GDELT GKG database. GDELT stores only metadata and does not contains the news articles contents, so, we can only be fetching the Arabic news URLs and the corresponding Average Tone values, from GDELT. The results of this SQL query are saved into CSV file format with two columns (news URL and average tone) and rows are equal to the total number of the obtained news. Fetching Arabic news URL from GDELT GKG database in three categories (positive, negative and neutral) is done using SQL query.

After acquiring a sufficient number of Arabic news metadata from GDELT, the next step is to obtain the contents of this Arabic news articles from the source URLs located in the CSV file that is previously obtained. For this task we utilized an open source article extraction tool called "Boilerpipe". When the number of the news which is required to be extracted becomes very large, the sequential extraction method which can be executed a piece at a time becomes inefficient and can be considered as time and compute intensive.

In order to address this issue we considered using a parallel article extraction method based on parallel computing. In this method, different extraction units can share the articles extracting task from different URLs and store the extracted article into text files simultaneously as shown in Fig. 1. In the multi-core processing environment, each one of these extraction unit processing tasks that can be assigned to different CPU cores and work independently. Since, our system contains a 32 core CPU, 32 extraction units are used to share the news article extraction tasks. Since the parallel extraction method can process and extract many articles in a shorter time compared to the ordinary sequential method that can reduce the extraction time and increase the performance. Figure 2 shows the time required to extract and store 100 news articles using a different number of extraction units.

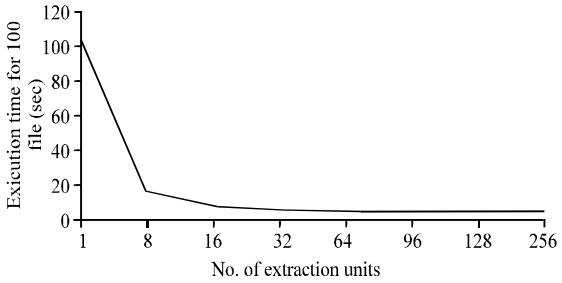

Fig. 2: News articals extraction time with respect to the number of paralell extraction units

Table 2: The total number of files in our GLASC corpus

Category	Final file number after filtering	Corpus size
Negative	266,376	816 MB
Positive	225,397	635 MB
Neutral	218,309	448 MB
Total	620,082	1.9 GB

Table 3: Statistics of our GLASC large-scale Arabic corpus

Category	Negative	Positive	Neutral
Number of files	266,376	225,628	218,310
Total number of words	91,051,658	70,596,129	51,061,595
The average number of words in each file	342	313	234
Total number of unique terms	155,929	154,336	156,752
Total number of unique terms in the corpus		230,123	
The average number of unique terms in each file	204	184	142
Total number of sentences	4,567,333	3,550,913	2,575,378
The average number of sentences in each file	17	16	12

The news articles contents that obtained in the previous step is stored and indexed with respects to its average tone values into three categories (Positive, negative and neutral). When all news contents text files are indexed and assigned to the positive, negative or neutral category then we applied filtering to remove the duplicated news text file and perform the final corpus.

The total number of files in our GLASC corpus which obtained from GDELT and the online Arabic news articles is shown in Table 2.

Corpus statistics and evaluation: We obtained different statistical measures related to the produced Arabic corpus such as the number of files in each category, the total number of words, the average number of words in each file, the total number of unique terms, the total number of unique terms in the corpus, the average number of unique terms in each file, the total number of sentences and the average number of sentences in each file as shown in Table 3.

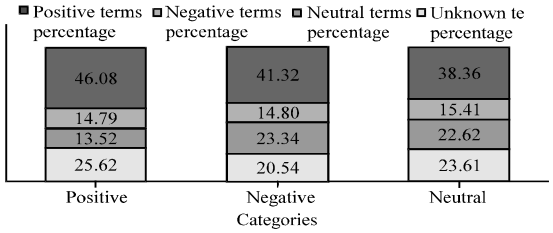

Fig. 3: The average number of the positive , negative and neutral terms in each catagory

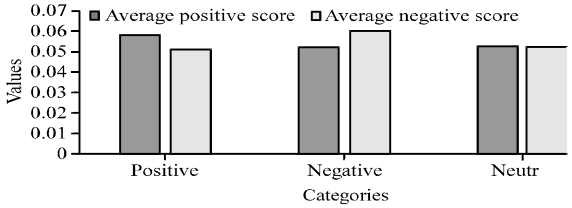

Fig. 4: The average positive and negative scores for the terms in each category

Since, we used tone value provided by GDELT to assign the news articles files into three various categories (positive, negative and neutral) to obtain our corpus, we need to evaluate the quality of this file assignment. For this task, we considered using ArSenL (Badaro *et al*., 2014) which is a large-scale standard Arabic sentiment and opinion-mining lexicon contains a total of 28,760 Arabic lemmas with corresponding sentiment scores.

In the first test we calculate the average number of the positive, negative and neutral terms in each category of our corpus using ArSenL as shown in Fig. 3.

The results shown in Fig. 3, verified that the positives, negative and neutral terms ratio in a specific category are compatible with the nature of that category, e.g. for the positive category the number of the positive terms is more than the number of negative and neutral terms.

The second test we performed over the corpus is calculating the average positive and negative scores for the terms in each category of our corpus using ArSenL as shown in Fig. 4.

The results are shown in Fig. 4, also confirm that the ratio of positive and negative terms scores is compatible with the category that contain these terms, e.g., in the positive category the positive terms scores are larger than scores of negative terms.

Machine learning approach for document-level Arabic sentiment analysis: In this study, we present our proposed ML approach for Arabic sentiment

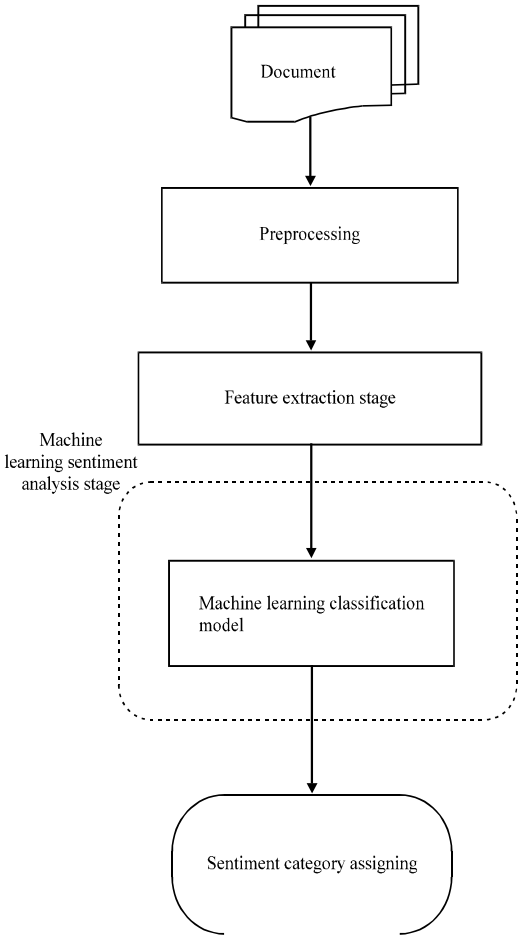

Fig. 5: The architecture of the proposed ML approach for document-level Arabic SA system

classification using our large-scale corpus. This approach evolves generating different training datasets using different feature extracting and feature weighting then using this dataset for train and evaluate various ML classifiers for sentiment analysis.

Datasets generation: We used our large-scale corpus for generating different datasets that will be used for training the ML classifier for sentiment analysis. Figure 5 shows the process of generating the datasets which consist of the following steps.

Prepressing: In this stage different text preprocessing techniques are applied to the document. These techniques include tokenizing the term in the document, removing the stop words and stemming the root of each term. For the root extraction task, we used the Buckwalter (2004) morphological analyzer's called Ara-morph Arabic lemmatizer.

Feature extraction and weighting: The second stage is the extraction and weighting, we used two different feature extraction methods unigrams and bigrams. In unigram method, each word or term in a document can be represented as a single feature where in bigram method each two adjacent words can be represented as a single feature.

Sentiment classification can be considered as a standard ML text categorization or document classification problem when each of the document can be represented by a fixed size numerical vector of terms or words. These vectors can be used as an input to the classification algorithm. Each term in the vector is typically weighted using the Term Frequency (TF) or the Term Frequency-Inverse Document Frequency (TF-IDF) (Sebastiani, 2002).

The TF score of a term is a value that indicates the frequency at which the term crosses the document. While there are many terms often found in many documents that are not trivial in terms of discretization it would be wrong to use these metrics in scoring. For this reason, IDF scores are derived. Here, the TF and IDF score for a specific term is calculated as:

$$TF(i,j) = \frac{\text{Number of times the term i appears in a document j}}{\text{Total number of terms in the document j}} \quad (1)$$

$$IDF(i) = \log\left(\frac{\text{Total number of document in the corpus}}{\text{Total number of document contain term i}}\right) \quad (2)$$

$$\text{TF-IDF}(i,j) = TF(i,j) \times IDF(j) \quad (3)$$

After calculating all the terms vectors for each document in the corpus, the dataset can be represented as a matrix where rows correspond to documents and columns to feature words (in our case TF and TF-IFD).

For our system evaluation, we generated four different datasets using two different feature extractions (unigrams and bigrams) and two different feature weighting methods (TF and TF-IDF). Table 4 provides a summary of each generated dataset properties.

In order to evaluate the sentiment classification models, each dataset is randomly divided into two different subsets which are training subset (used to train all the classifiers models) and a testing subset (used to evaluate the classifiers models). Training subset contains 70% of the total dataset instances when test subset contains 30%.

Table 4: The properties of the generated datasets

Types	Dataset-1	Dataset-2	Dataset-3	Dataset-4
Features type	Unigrams	Unigrams	Bigrams	Bigrams
Features weighting	TF	TF-IDF	TF	TF-IDF
Number of features	230,123	230,123	5,600,000	5,600,000
Train instances	Negative	Positive	Neutral	
	186,463	157,940	152,817	
Test instances	Negative	Positive	Neutral	
	79,913	67,688	4,5845	

Machine learning sentiment classifier: A classifier is a ML approach that places data items into one of C classes based on previous knowledge. The major goal of the classification algorithm is to maximize the classification accuracy with instances that are not included in the training set (Sebastiani, 2002).

In our experiments, we used Support Vector Machine (SVM) (Joachims, 1998), Hidden Markov Model (HMM) (Soni and Sharaff, 2015), Neave Bayes (NB) (Dhande and Patnaik, 2014), Artificial Neural Network (ANN) (Sharma and Dey, 2012) and K-Nearest Neighbors (KNN) (Jiang *et al*., 2012) classification algorithms which are widely used in sentiment analysis and can be considered as a baseline benchmark for any further experiments on the datasets.

In ensemble learning a multiple ML-based models are cooperatively works together for solving the same problem. An ensemble classifier combines the decisions of the individual weak classifiers and aims to enhance the accuracy final decision and produce a stronger classifier. There are basically 2 approaches for combining classifiers, one approach is to use similar classifiers and to combine them together using techniques such as Bagging, boosting or random subset. A second approach is to combine different classifiers using model fusion using stacking technique (Wang *et al*., 2014).

Bagging: In this method, different training sets are used for training multiple classifier models from the same type. A method based on sampling and replacement is applied for creating the multiple training sets that used in bagging method. The decision of classifying an unknown instance is done with respect to the majority voting of all results that obtained by the ensemble classifier models (Su *et al*., 2012).

Boosting: In this method, different training sets with weighted instances are used for training multiple classifier models from the same type sequentially. This method focuses on the training samples that misclassified by the previous classifiers in the chain by using higher weights to the misclassified instance before passing it to the next classifier. The final decision is obtained by combining decisions of base classifiers by a voting scheme (Su *et al*., 2012).

Random subspace: This method is similar to bagging but the difference that it is selects a random subset of features from the dataset instead of instances. In random subspace, different training sets with different features subspaces are used for training multiple classifier models from the same type. If there are many of irrelevant and redundant features in training dataset, so, using random subspaces may results in overcoming these unwanted features, since it creates multiple training sets with different features subspaces drawn randomly from the original dataset. Similar to the other ensemble methods the final decision is obtained by combining decisions of base classifiers by a voting scheme (Su *et al*., 2012).

Stacking: Staking is a technique that fuses multiple classifiers applied to a specific classification problem and aims to improve the results of the individual classifier (Ruta and Gabrys, 2000). Staking method combines multiple classifier models from different types using another classifier called meta-classifier in a stacked structure. The meta-classifier is trained based on the output of each combined model using staking.

The classification task in stacking is achieved in two stages. In the first stage each one of combined model generates a classification decision for the unknown instance, then is the second stage these output decisions are fed as input to the meta-classifier which is, in turn, provides the final classification decision for the unknown instance (Dzeroski and Zenko, 2004; Ruta and Gabrys, 2000).

RESULTS AND DISCUSSION

To evaluate the performance of the sentiment classification model we considered using 10-fold cross-validation method to calculate the model accuracy and F-score which can be calculated as following (Sebastiani, 2002):

$$\text{Accuracy} = \frac{1}{c}\sum_{i=1}^{c}\frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i} \quad (4)$$

$$\text{Precision} = \frac{1}{c}\sum_{i=1}^{c}\frac{TP_i}{TP_i+FP_i} \quad (5)$$

$$\text{Recall} = \frac{1}{c}\sum_{i=1}^{c}\frac{TP_i}{TP_i+FN_i} \quad (6)$$

$$\text{F-score} = \frac{2\times\text{recall}\times\text{precision}}{\text{Recall}+\text{precision}} \quad (7)$$

where, c refers to the number of classes and True Positive (TP) and True Negative (TN) refers to the number of instances that correctly classified by the model, False Positive (FP) and False Negative (FN) refers to the number of instances that miss-classified by the model.

In the first experiment we used the four sentiment datasets generated earlier individually for training the five base learners (SVM, HMM, NB, NN and KNN). The 10-fold cross validation method was considered to reduce the influence of variability in the training dataset. So that each individual classification model is trained on 70% of randomly drawn samples from the original dataset and tested on remaining 30% samples repeatedly for ten times and each time when testing dataset is applied the classification accuracy and F-score performance metrics are calculated. At the end of the 10-folds, the average value of the calculated accuracy and F-score performance metrics are obtained. This process is applied to each individual base learner using each one of the four datasets separately. The obtained values of the accuracy and F-score performance metrics for each base learner using four different datasets are shown in Table 5.

According to our experiments for the all the four datasets the best classification performance is achieved by both of SVM and HMM classification methods. SVM classifier provided its best performance of 87.76% F-score by using the dataset with bigram features and TF-IDF weights followed by HMM classifier provided its best performance of 86.75% F-score using also the same dataset.

In addition to the base classifiers evaluation in the previous experiment we consider another set of experiments to evaluate ensemble classifiers with the same 4 datasets and evaluation metrics.

We used three different classifier model ensemble methods which are bagging, boosting, random subspace and stacking. These methods used to combine the same type of classifiers models for each one of (SVM, HMM, NB, ANN and KNN) base learners in order to increase the classification performance.

Table 5: The classification accuracy and f-score for base learners using four different datasets

Factors	Dataset-1	Dataset-2	Dataset-3	Dataset-4
SVM				
Accuracy	0.8525	0.8547	0.8699	0.8776
F-score	0.8491	0.8527	0.8662	0.8706
HMM				
Accuracy	0.8271	0.8457	0.8692	0.8675
F-score	0.8183	0.8352	0.8629	0.8662
NB				
Accuracy	0.7730	0.7755	0.7838	0.8008
F-score	0.7726	0.7682	0.7832	0.7905
ANN				
Accuracy	0.7212	0.7229	0.7375	0.7407
F-score	0.6725	0.7226	0.7284	0.7396
KNN				
Accuracy	0.5503	0.5655	0.6501	0.6671
F-score	0.4725	0.5128	0.5716	0.6109

Table 6: The classification Accuracy and F-score for ensemble learners using Bagging method for four different datasets

Factors	Dataset-1	Dataset-2	Dataset-3	Dataset-4
SVM				
Accuracy	0.8595	0.8578	0.8724	0.8839
F-score	0.8574	0.8548	0.8688	0.8793
HMM				
Accuracy	0.8332	0.8418	0.8731	0.8740
F-score	0.8254	0.8290	0.8679	0.8733
NB				
Accuracy	0.7634	0.7753	0.7863	0.8137
F-score	0.7629	0.7668	0.7851	0.8072
ANN				
Accuracy	0.7253	0.7340	0.7437	0.7486
F-score	0.6804	0.7335	0.7330	0.7476
KNN				
Accuracy	0.5569	0.5562	0.6621	0.6701
F-score	0.5037	0.5124	0.5936	0.6175

Bagging method generates 5 different bootstrap training subsets which are drawn from the original training set with replacement. These five training subsets are used to train five model form similar type of classifier for each of (SVM, HMM, NB, ANN and KNN). The final prediction is generated by the majority voting of these five models.

Table 6 shows the accuracy and F-score results for each ensemble classifier using bagging method after training and evaluating with our four sentiment datasets using 10-fold cross-validation.

Boosting used to train five classification model form similar type of classifier for each of (SVM, HMM, NB, NN and KNN) in sequence. Each classification model is focused on the misclassified samples by the preceding model. Similar to bagging the final classification decision made by majority voting of these five models. The accuracy and F-score results of each ensemble classifier using boosting method after training and evaluation using our four sentiment datasets are shown in Table 7.

Table 7: The classification accuracy and F-score for ensemble learners using Boosting method for four different datasets

Factors	Dataset-1	Dataset-2	Dataset-3	Dataset-4
SVM				
Accuracy	0.8578	0.8330	0.8876	0.8837
F-score	0.8548	0.8285	0.8821	0.8765
HMM				
Accuracy	0.8418	0.8194	0.8771	0.8799
F-score	0.8290	0.8138	0.8698	0.8793
NB				
Accuracy	0.7753	0.7754	0.7952	0.8013
F-score	0.7668	0.7745	0.7944	0.7930
ANN				
Accuracy	0.7340	0.7305	0.7398	0.7443
F-score	0.7335	0.6820	0.7338	0.7436
KNN				
Accuracy	0.5562	0.5551	0.6532	0.6730
F-score	0.5124	0.5047	0.5854	0.6133

Table 8: The classification accuracy and F-score for ensemble learners using random subspace method for four different datasets

Factors	Dataset-1	Dataset-2	Dataset-3	Dataset-4
SVM				
Accuracy	0.8605	0.8692	0.8907	0.9057
F-score	0.8578	0.8675	0.8858	0.9021
HMM				
Accuracy	0.8464	0.8515	0.8898	0.8926
F-score	0.8354	0.8391	0.8839	0.8921
NB				
Accuracy	0.7852	0.7868	0.7788	0.8070
F-score	0.7843	0.7821	0.7783	0.7992
ANN				
Accuracy	0.7185	0.7404	0.7459	0.7520
F-score	0.6775	0.7399	0.7395	0.7511
KNN				
Accuracy	0.5674	0.5814	0.6635	0.6884
F-score	0.5185	0.5310	0.5992	0.6376

Random subspace method is similar to bagging method in concept, however, it's trained five similar models for each classification method on the same dataset with random subspaces when each random subspace contains 50% of the available feature space. Table 8 shows the accuracy and F-score results for each ensemble classifier using random subspace method after training and evaluating with our four sentiment datasets with 10-fold cross-validation.

The experimental results for (Bagging, boosting and random subspace) ensemble methods, shown in Table 6-8 can be summarized as follows: the classification accuracy results achieved by bagging, boosting and random subspace ensemble methods are higher than classification accuracy results achieved by the based learners. for all ensemble methods, the best classification accuracy is achieved by SVM and HMM classifiers using the dataset with bigram features and TF-IDF weights. Random subspace ensemble method achieved the highest classification accuracy over the other bagging and boosting ensemble methods. The best explanation for this phenomenon is that most of the learning algorithms are sensitive to the dimensionality of the training data in a negative manner and since, sentiment classification problem has a high dimensional feature space data that may contain noisy features which may lead to overfitting problem. Since, random subspace ensemble is based on feature partitioning, so it can reduce the risk of overfitting problem and improve the classification performance.

Table 9: The classification accuracy and F-score for combined learners using stacking method for four different datasets

Factors	Dataset-1	Dataset-2	Dataset-3	Dataset-4
SVM+NB				
Accuracy	0.8633	0.8599	0.8811	0.9144
F-score	0.8525	0.8599	0.8775	0.9142
NB+HMM				
Accuracy	0.8263	0.8515	0.8794	0.8851
F-score	0.8184	0.8392	0.8779	0.8851
ANN+SVM				
Accuracy	0.8589	0.8651	0.8750	0.9071
F-score	0.8517	0.8582	0.8733	0.9057
SVM+HMM				
Accuracy	0.8662	0.8863	0.8974	0.9238
F-score	0.8661	0.8839	0.8967	0.9235

Stacking is a classification model fusion method which is concerned with combining multiple classifiers generated by using different learning on a single dataset. This method implies two stages, the first stage consists of training different classification models called base-level classifiers. In the second stage, a meta-level classifier is learned that combines the outputs of the base-level classifiers and the predictions of base learners (level-0 models) are used as input for meta-learner (level-1 model).

We performed different batches of experiments, wherein each experiment we used a different combination of classifiers methods such as (SVM+NB, NB+HMM, NN+SVM and SVM+HMM) as level-0 models. For the level-1 model, we used a multilayer perceptron MLP as meta-classifier to combine the decisions of the level-0 classifier models. Table 9 shows the accuracy and F-score results for each classifier combination using the stacking method after applying 10-fold cross validation for training and evaluating with our four sentiment datasets.

The results in Table 9 can show that by using stacking method we were able to improve the accuracy of the all combined (fused) classification models rather than using the induvial models. The highest classification accuracy of 92.35% F-score is achieved by SVM+HMM classifiers fusion, followed by 91.42% F-score by SVM+NB classifiers fusion, using the dataset with bigram features and TF-IDF weights. The summary of the obtained experimental result is shown as follows:

The datasets with bigrams features produce classification models with higher accuracy than the datasets with unigrams features. Using TF-IDF rather than TF feature weighting can provide an enhancement in classification performance. The maximum classification

performance is provided by SVM base learner classifier (which has been proven by many of previous research that SVM has the more powerful competitiveness in text classification application especially sentiment classification (Abdulla *et al*., 2013; Aly and Atiya, 2013; Korayem *et al*., 2012; Mullen and Collier, 2004; Saleh *et al*., 2011). In general, SA can be considered as text classification problem with linearly separable categories and, since SVM classification always assumes a hyperplane exist between the classes/categories, so it performs better when the classes/categories are linearly separable as in text classification problem. Also when the number of data dimensions is very high SVM can be superior to the other classification method in performance wise. In classification performance wise, after SVM classifier the HMM classification method takes the second place followed by NB, NN and KNN.

In general, classification method such as SVM, HMM and NN performs better when it deals with higher dimensional data, however, ANN is more prone to suffer from multiple local minima and overfitting issues which can reduce the performance. On the other hand, classification method such as NB and KNN provide a better performance when working with lower dimensional data (Abdulla *et al*., 2013; Mountassir *et al*., 2012; Saleh *et al*., 2011; Shoukry and Rafea, 2012). KNN classifier achieved the worst classification performance among the all other classifiers (KNN classification algorithm uses the Euclidean distance between the data point to classify a new unknown instance and since the datasets used for sentiment classification tends to have higher dimensionality, this distance measure becomes meaningless and can reduce the classification performance in general (Jedrzejewski and Zamorski, 2013; Sebastiani, 2002).

Another fact that can affect the classification performance is using imbalanced dataset where the numbers of (positive, negative and neutral) samples that used to train the ML model are not equal. The results show the effectiveness of using ensemble learning methods (bagging, boosting and random subspace and staking) in term of improving the classification performance. The best classification performance is achieved by using random subspace ensemble method (which combine similar type of classifier models) and staking classifier fusion method (which combine different type of classifier models).

Using classifier model fusion by stacking method can improve the performance in term of accuracy of the all combined (fused) classification models rather than using each single classifier model separately. The maximum classification performance is achieved by using SVM+HMM classifier fusion model with the highest F-score value of 92.35%, so that, this method is considered for a generation the sentiment classification model that used in our proposed document-level Arabic SA system.

CONCLUSION

In this research we presented a large-scale Arabic sentiment analysis corpus called GLASC which built using online Arabic news articles and metadata provided by the bigdata resource GDELT. Our corpus consists of a total of 620,082 Arabic news articles divided into three categories (Positive, negative and neutral). Besides that our corpus also provides a sentiment rating by assigning a sentiment score in a range between-1 and 1 for each article. We carried out two different types of experiments in order to evaluate the quality of the generated GLASC corpus. The first evaluation experiment involves using statistical measures to calculate the percentage of positive, negative and neutral terms in each positive, negative and neutral category in our corpus based on Arabic Sentiment Lexicon called (ArSenL). The second evaluation experiment involves comparing the term rank to term frequency distribution of our GLASC corpus to the ideal Zipf distribution. To our best knowledge, this corpus can consider as the largest resource available for Arabic language and we believe it will provide a significant contribution not only to sentiment analysis but a wide range of Arabic NLP applications in Big Data domain. We used our GLASC corpus to build an Arabic document-level SA system based on ML classification and regression approaches when an ML-based classifier model is used for assigning an Arabic document into one of three various categories (Positive, negative or neutral) and a ML-based regression model used for predicting the sentiment score of the Arabic document based on its sentiment orientation.

We have also carried out some experiments on our corpus, using ML algorithms to generate sentiment classifier. For training the sentiment classifier we generated four datasets from our corpus using different feature extraction and feature weighting method. We performed a comparative study, involving testing a wide range of classifiers that commonly used for sentiment analysis task such as (SVM, HMM, NB, NN and KNN).

In additional we investigated several types of ensemble learning methods such (Bagging, boosting, random subspace and staking) to verify its impact of improving the classification performance for sentiment analysis, using different comprehensive empirical experiments.

For all experiments done, the best classification performance is achieved using a dataset with Bigram features and TF-IDF weights over the other three datasets.

The obtained results showed that as a base learner SVM and HMM have achieved the best results with an F-score of 87.06% for SVM and with an f-score of 86.75% for HMM.

Our experiments result also verified the effectiveness of using ensemble learning methods (bagging, boosting and random subspace and staking) in term of improving the classification performance. The ensemble model of SVM using random subspace method achieved the best classification accuracy of 90.21% of an F-score and the ensemble model of HMM using the same method achieved an f-score of 89.21%.

With regard to the results of our experiments, the maximum classification performance is achieved by using stacking classifier fusion method with the highest value of 92.35% of F-score for the SVM+HMM classifiers fusion and a value of 91.42% of F-score for the SVM+NB classifiers fusion.

REFERENCES

Abdul-Mageed, M. and M.T. Diab, 2012. AWATIF: A multi-genre corpus for modern standard Arabic subjectivity and sentiment analysis. Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'14), May 26-31, 2014, European Language Resources Association, Reykjavik, Iceland, pp: 3907-3914.

Abdul-Mageed, M., M. Diab and S. Kubler, 2014. SAMAR: Subjectivity and sentiment analysis for Arabic social media. Comput. Speech Lang., 28: 20-37.

Abdulla, N.A., N.A. Ahmad, M.A. Shehab and M. Al-Ayyoub, 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. Proceedings of the IEEE Conference on Applied Electrical Engineering and Computing Technologies (AEECT'13), December 3-5, 2013, IEEE, Amman, Jordan, ISBN:978-1-4799-3676-2, pp: 1-6.

Al-Smadi, M., O. Qawasmeh, B. Talafha and M. Quwaider, 2015. Human annotated Arabic dataset of book reviews for aspect based sentiment analysis. Proceedings of the 3rd International Conference on Future Internet of Things and Cloud (FiCloud'15), August 24-26, 2015, IEEE, Rome, Italy, ISBN:978-1-4673-8103-1, pp: 726-730.

Aly, M. and A. Atiya, 2013. Labr: A large scale Arabic book reviews dataset. Proceedings of the 51st Annual Meeting on Association for Computational Linguistics, August 4-9, 2013, Association for Computational Linguistics, Sofia, Bulgaria, pp: 494-498.

Anonymous, 2015. GDELT translingual: Translating the planet. GDELT Project, Global Database of Events, Language and Tone (GDELT), Georgetown, USA. https://blog.gdeltproject.org/gdelt-translingual-translating-the-planet/

Baayen, R.H., 2001. Word Frequency Distributions. Kluwer Academic Publishers, Dordrecht, the Netherlands, Pages: 341.

Badaro, G., R. Baly, H. Hajj, N. Habash and W. El-Hajj, 2014. A large scale Arabic sentiment lexicon for Arabic opinion mining. Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP'14), October 25, 2014, Association for Computational Linguistics, Doha, Qatar, pp: 165-173.

Buckwalter, T., 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, Philadelphia, Pennsylvania, USA., ISBN:1-58563-324-0,.

Dave, K., S. Lawrence and D.M. Pennock, 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. Proceedings of the 12th International Conference on World Wide Web, May 20-24, 2003, ACM, Budapest, Hungary, pp: 1-10.

Dhande, L.L. and G.K. Patnaik, 2014. Analyzing sentiment of movie review data using Naive Bayes neural classifier. Intl. J. Emerging Trends Technol. Comput. Sci., 3: 313-320.

Dzeroski, S. and B. Zenko, 2004. Is combining classifiers with stacking better than selecting the best one?. Mach. Learn., 54: 255-273.

El-Sahar, H. and S.R. El-Beltagy, 2015. Building large Arabic multi-domain resources for sentiment analysis. Proceedings of the 16th International Conference on Intelligent Text Processing and Computational Linguistics, April 14-20, 2015, Springer, Cairo, Egypt, ISBN:978-3-319-18116-5, pp: 23-34.

Elliot, C., 1992. The affective reasoner: A process model of emotions in a multi-agent system. Master Thesis, Northwestern University, Evanston, Illinois.

Gandomi, A. and M. Haider, 2015. Beyond the hype: Big data concepts, methods and analytics. Int. J. Inform. Manage., 35: 137-144.

Jedrzejewski, K. and M. Zamorski, 2013. Performance of K-nearest neighbors algorithm in opinion classification. Found. Comput. Decis. Sci., 38: 97-110.

Jiang, S., G. Pang, M. Wu and L. Kuang, 2012. An improved k-nearest-neighbor algorithm for text categorization. Expert Syst. Appl., 39: 1503-1509.

Joachims, T., 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Machine Learning, Nedellec, C. and C. Rouveirol (Eds.). Springer, Berlin, Germany, ISBN:978-3-540-64417-0, pp: 137-142.

Korayem, M., D. Crandall and M. Abdul-Mageed, 2012. Subjectivity and Sentiment Analysis of Arabic: A Survey. In: Advanced Machine Learning Technologies and Applications, Hassanien, A.E., M.S. Abdel-Badeeh, R. Ramadan, and K.T. Hoon (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-35325-3, pp: 128-139.

Mountassir, A., H. Benbrahim and I. Berrada, 2012. Some methods to address the problem of unbalanced sentiment classification in an Arabic context. Proceedings of the 2012 Conference on Colloquium in Information Science and Technology (CIST'12), October 22-24, 2012, IEEE, Fez, Morocco, ISBN:978-1-4673-2726-8, pp: 43-48.

Mullen, T. and N. Collier, 2004. Sentiment analysis using support vector machines with diverse information sources. Proceedings of the Conference on Empirical Methods on Natural Language Processing, July 25-26, Association for Computational Linguistics Press, Barcelona, Spain, pp: 412-418.

Nasukawa, T. and J. Yi, 2003. Sentiment analysis: Capturing favorability using natural language processing. Proceedings of the 2nd International Conference on Knowledge Capture, October 23-25, 2003, ACM, Sanibel Island, Florida, USA., ISBN:1-58113-583-1, pp: 70-77.

Ortony, A. and T.J. Turner, 1990. What's basic about basic emotions?. Psychol. Rev., 97: 315-331.

Pang, B. and L. Lee, 2008. Opinion mining and sentiment analysis. Found. Trends Inform. Retrieval, 2: 1-135.

Pang, B., L. Lee and S. Vaithyanathan, 2002. Thumbs up?: Sentiment classification using machine learning techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing Vol. 10, July 6-7, 2002, Association for Computational Linguistics, Stroudsburg, Pennsylvania, pp: 79-86

Ptaszynski, M., R. Rzepka, S. Oyama, M. Kurihara and K. Araki, 2014. A survey on large scale corpora and emotion corpora. Inf. Media Technol., 9: 429-445.

Ruta, D. and B. Gabrys, 2000. An overview of classifier fusion methods. Comput. Inf. Syst., 7: 1-10.

Sagi, D.J.B. and J.M. Labeaga, 2016. Using GDELT data to evaluate the Con dence on the Spanish government energy policy. Intl. J. Interact. Multimedia Artif. Intell., 3: 38-43.

Saleh, M.R., M.T.M. Valdivia, L.A.U. Lopez and J.M.P. Ortega, 2011. OCA: Opinion corpus for Arabic. J. Am. Soc. Inf. Sci. Technol., 62: 2045-2054.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Comput. Surveys, 34: 1-47.

Sharma, A. and S. Dey, 2012. An artificial neural network based approach for sentiment analysis of opinionated text. Proceedings of the 2012 ACM Symposium on Research in Applied Computation, October 23-26, 2012, ACM, San Antonio, Texas, ISBN:978-1-4503-1492-3, pp: 37-42.

Shoukry, A. and A. Rafea, 2012. Sentence-level Arabic sentiment analysis. Proceedings of the 2012 International Conference on Collaboration Technologies and Systems (CTS'12), May 21-25, 2012, IEEE, Denver, Colorado, ISBN:978-1-4673-1381-0, pp: 546-550.

Soni, S. and A. Sharaff, 2015. Sentiment analysis of customer reviews based on hidden markov model. Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering and Technology (ICARCSET'15), March 06-07, 2015, ACM, Unnao, India, ISBN:978-1-4503-3441-9, pp: 1-5.

Su, Y., Y. Zhang, D. Ji, Y. Wang and H. Wu, 2012. Ensemble learning for sentiment classification. Proceedings of the 13th Workshop on Chinese Lexical Semantics, July 6-8, 2012, Springer, Wuhan, China, ISBN:978-3-642-36336-8, pp: 84-93.

Taboada, M., J. Brooke, M. Tofiloski, K. Voll and M. Stede, 2011. Lexicon-based methods for sentiment analysis. Comput. Ling., 37: 267-307.

Wang, G., J. Sun, J. Ma, K. Xu and J. Gu, 2014. Sentiment classification: The contribution of ensemble learning. Decis. Support Syst., 57: 77-93.

Zipf, G.K., 1935. The Psycho-Biology of Language. Houghton Mifflin Harcourt, Oxford, England, UK.,.