Building a Sentiment Analysis System Using Automatically Generated Training Dataset

https://doi.org/10.3991/ijoe.v16i06.13623

Daoud M. Daoud (✉)
Higher Colleges for Technology, Sharjah, UAE
ddaoud@hct.ac.ae

M. Samir Abou El-Seoud
The British University, Cairo, Egypt
Higher Colleges for Technology, Sharjah, UAE

Abstract—In this paper, we describe a methodology to develop a large training set for sentiment analysis automatically. We extract Arabic tweets and then annotates them for negativeness and positiveness sentiment without human intervention. These annotated tweets are used as a training data set to build our experimental sentiment analysis by using Naive Bayes algorithm and TF-IDF enhancement. The large size of training data for a highly inflected language is necessary to compensate for the sparseness nature of such languages. We present our techniques and explain our experimental system. We use 200 thousand annotated tweets to train our system. The evaluation shows that our sentiment analysis system has high precision and accuracy measures compared to existing ones.

Keywords—Sentiment Analysis; Arabic; Naive Bayes; TF-IDF weight scheme.

1 Introduction

Opinions are difficult to extract and search using the current information retrieval relevancy methods. It is even more challenging for languages with little resources such as Arabic.

The significance of sentiment analysis is increasing because people tend to express their sentiments on different things and topics. The widespread usage of social media channels accelerates this phenomenon.

For every business, it is important to collect comments about their services and products. All social media channels, such as tweets and reviews, are a rich source of opinions. Thus, sentiment analysis is becoming an important mean which helps by automating the process of extracting the opinions from diverse content channels. Alongside, companies, the political parties, the government are also more and more interested in this type of analysis to extract opinion polarity from tweets, Facebook messages, and blogs.

Twitter is now one of the most widespread networks for exchanging information among Arab people. The use of micro-blogging exploded during the current events as people turned to social media channels to express themselves.

Analyzing some Arabic tweets reveals that the text holds many different surface structures for the same meaning [1].

The variety of surface structures reflects the diversity of the posters and the richness of the Arabic language itself [2]. It was obvious from studying the tweets that there was no unified syntactic structure that is widely used by all the posters. We observe that some posts contain fragmented expressions (telegraphic) rather than full sentences. Other tweets are more consistent, and some are full sentences. Extensive linguistic-based methods would not prove very useful in dealing with the given posts. For example, it is not practical to analyze a tweet by finding the object and subject as we do when we analyze correct Arabic sentences. Likewise, methods used for semi-structured text by utilizing position, layout, and set-up of text are not appropriate.

We also observe the flexible order nature of the tweets, which is an expression of the enormously free ordered nature of the Arabic Language. Additionally, we find that many posts are not written in Modern Standard Arabic. We find many posts are written in different dialects.

Because of the above reasons, we avoid using fully featured linguistic processing methods to extract opinion from the text. The other option is to use machine learning and statistical models.

However, adopting a machine learning approach requires a sizeable Arabic training set. To handle the richness of Arabic and its inflection nature, we need a training set consisting of thousands of tweets that are rich with negative and positive terms. Such a training dataset does not exist [3, 4].

In this paper, we propose an innovative approach to collect and label positive and negative polarity data without human intervention. The source of the data is posted from Twitter. Our strategy is to keep the linguistic processing minimum. We use a sophisticated stemming algorithm developed by the first author. Additionally, we use some information retrieval technique to accomplish this task.

Then we use the vector space model and calculate the weights using TF-IDF scheme. These vectors are fed to Naive Bayes algorithm to build the training model.

The outline of this paper is as follows: in the next section, we will present the related work. Then, we present the proposed solution. After that, we will explain the methods used to build the training data. Then, we describe the overall architecture of the system. Finally, we present the results of the evaluation of our sentiment analysis system.

2 Related Work

Sentiment Analysis is usually carried out using three typical approaches: supervised machine learning, unsupervised approach, and hybrid approach [5-7]. In a supervised approach, a large labeled data set is required to train the classifier. For the Arabic language building, a data set is time-consuming and requires a lot of resources[8]. The dominating training algorithm that is used in this approach is Support Vector Machine (SVM), Naïve Bayes (NB) [9].

The unsupervised approach does not require training data set. The generated clusters need to be labeled correctly using a large lexicon. The hybrid tries to be pragmatic and uses both approaches [10, 11].

Concerning Arabic sentiment analysis, machine learning techniques are used more than other approaches such as Lexicon based and hybrid techniques.

Some studies indicated that Naive Bayes and SVM algorithms outperform other algorithms. Some papers reported high performances: accuracy (96.06%), precision (95.80%) and recall 96.40%) [3].

Concerning the training datasets, Arabic lacks open resources. Thus, each research group build their resources. Researchers report their results on their dataset with no possibility for benchmarking results and assessing the experiments. Some of the work on sentiment analysis is using MSA text. Other researchers have recently processing text collected from social media and designed to handle both Arabic dialects as well as MSA text.

In this experiment, we will use tweets as a source for training and testing our system. We will combine light linguistic processing to enhance the results combined with TF-IDF (Term Frequency-Inverse Document Frequency). The main contribution of this research is the automatically labeled training set.

3 Proposed Model

In our model, we use Naive Bayes with TF-IDF to build our training model, as shown in figure 1. The extracted tweets are filtered, and only Arabic tweets are kept. We also remove embedded links and non-Arabic hashtags from each tweet. Generally, we keep only Arabic letters in these tweets. By the end of this phase, the training data file is generated. Each line of the training set file contains the category (1 for positive and 0 for negative) and the tweet body.

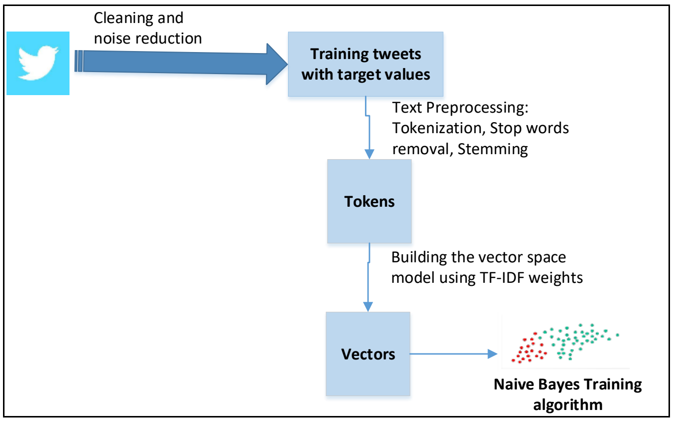
Cleaning and noise reduction
Training tweets with target values
Text Preprocessing: Tokenization, Stop words removal, Stemming
Tokens
Building the vector space model using TF-IDF weights
Vectors
Naive Bayes Training algorithm

Fig. 1. Steps and techniques used in our model

3.1 Text preprocessing

Preprocessing of text is required to enhance and improve the results. The reduction of noise in the text should help increase the performance of the sentiment analysis and opinion extraction system.

1. Tokenization: For a given input as a string, tokenization is a task of breaking up it up into small, usable chunks of text, called tokens. This process involves removing certain characters, such as punctuation marks. Tokens often represent single words and considered a useful semantic unit for processing. The most direct approach to tokenizing Arabic is to split up a sequence of characters based on the occurrence of whitespaces, such as spaces and line breaks.
2. Stop words elimination: A stop-list is the name usually given to a set or list of stop words. It is usually language specific, although it may contain words. A search engine or other natural language processing system may contain a variety of stop-lists, one per language, or it may contain a single stop-list that is multilingual. Some of the more frequently used stop words for Arabic include "من", "في", "الى", "هو", "هي", " على"", and” منك” these are generally regarded as 'functional words' which do not carry meaning. When assessing the contents of natural language, the meaning can be conveyed more clearly by ignoring the functional words. Hence it is practical to remove those words which appear too often that support no information for the task.
3. Stemming: Detecting the surface variations of the same word is one of the main challenges of any natural language processing system. Specifically, the effectiveness for information retrieval depends on its ability to map all those variations to the same form.

Stemming is the process of automatically revealing a word‘s stem. In other words, stemming a word is the removal of all the inflectional morphemes from the word's surface-form. Lemmatization goes a step further in identifying the citation form of the word, also often called its lemma, typically used to access dictionaries. In many languages, the inflected or derived word forms of a lemma have several stems.

Arabic is a Semitic language, and its main feature is the rich morphology in which most of its words are originated from roots. Inflections and derivations are produced by changing vowels and addition of consonants.

Arabic text is characterized by a strict and obvious agreement between its elements, between verb and noun, noun and adjective, in matters of numbers, gender, definitiveness, case, person, etc. These attributes are expressed by a wide-ranging structure of affixation and inner inflections. Arabic uses a diverse system of prefixes, suffixes, and pronouns that are connected to the words, producing compound forms that further complicate text processing (Daoud 2005). For example, articles such as "the" is not a distinct word as it is in languages like English but are attached to the words to which they refer (for example, "their two books" is written as a single word.

Thus in Arabic, any given word will appear lesser amount than in English. In other words, an Arabic collection of text will have a greater level of sparseness compared to English.

3.2 Transformation and building the vector space model

The tokens extracted from corpus are considered as attributes of that text. These tokens, which are treated as features, are used in classification algorithms to decide the class or category of the documents. The simplest method, known as the bag-of-words approach, treats a document as a set of words. Each word appearing in a tweet is considered a feature or attribute, and these features are calculated according to their number of occurrence (term frequency).

On the other hand, the term frequency-inverse document frequency (TF-IDF) weighting scheme is used to generate values for each term, in order to assign importance to words in a document based on how frequently each term occurs in the whole training corpus. Depending upon the size of the corpus, it may be necessary to use a subset of the terms in the corpus as the set of features to build a classifier upon. Eliminating words that occur frequently or that have a low IDF allows to train a classifier on the most important words in the corpus: the words that have the greatest strength in discriminating between different categories.

To explain weighting scheme, say that a document has tokens t1, t2, ..., tn with frequencies f1, f2, …, fn. The term frequency (TFi) of token ti is the frequency fi.

To compute the inverse document frequency, the document frequency (DF) for each word is first calculated. Document frequency is the number of documents the term found in. Then, the inverse document frequency or IDFi for a term, ti, is

$$IDF_i = \frac{N}{DF_i}$$

Where N is the document count.

Thus, the TF-IDF weight Wi of a term ti in a document vector is

$$W_i \;=\; TF_i . \log\frac{N}{DF_i}$$

The TF-IDF transformation will be used before applying the learning algorithm.

3.3 Naive Bayes training algorithm

In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes’ theorem with strong (naive) independence assumptions between the features. The training process involves analyses of the relations between tokens in the training documents and categories (1 for positive and 0 for negative), and the associations between categories and the whole training set. The facts are assembled using calculations based on Bayes’ Theorem to generate the probability that a group of words (a document) belongs to a precise classification.

During the training process, the naive Bayes algorithm finds out the frequency each word appears in a document in a given class and divides that by the number of words appearing in that class.

This is referred to as a conditional probability; in this case, the probability that a word will appear in a particular category. This is commonly written as P(Word | Category).

In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)},$$

where A and B are events and P(B) # 0.

P(A) and P(B) are the probabilities of A and B without regard to each other.

P(A | B), a conditional probability, is the probability of observing event A given that B is true.

P(B | A) is the probability of observing event B given that A is true.

Each text we will classify contains words noted with Wi (i=1..n). For each word Wi from the training data set we can extract the following probabilities (noted with P):

4 Preparing the Training Dataset

Sentiment classification at both the document and sentence levels are considered useful, but they do not find what the opinion holder exactly liked and disliked. For example, a negative sentiment on an object does not mean that the opinion holder dislikes everything about the object. Also, a positive sentiment on an object does not mean that the opinion holder likes everything about the object. So we need to go to the feature-level and classify the sentiments there based on what the opinion holder likes and dislikes of the features of this object or that one. However, before we dig any deeper, let us discuss how the opinions are formed of words and phrases.

The Arabic Language lacks many resources that are freely available to be used by researchers and developers. It cannot be compared with resource-rich languages like English. To conduct this experiment, an extensive training set is required. When we say an extensive training set, we assume it is to be above 50 thousand annotated sentences. This size of data is only available on social media channels. Twitter is suitable for this task: it has a limited size of post length, written by different groups and people and it has the APIs that facilitate collecting a considerable number of tweets as shown in Figure 2.

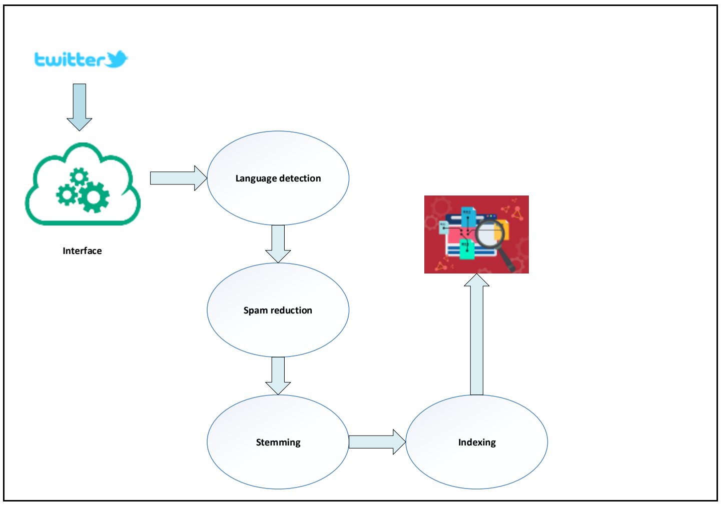

Fig. 2. The architecture of data collection component

Since we are interested in Arabic only, we need to detect and accept only tweets that contain at least six Arabic characters. We also need to reduce the noisy tweets, which are bots. We have noticed that many of the tweets are generated automatically. Their contents vary from religious, governmental propaganda, commercial ads, etc... The spam detection component rejects/accepts tweets based on different conditions: hashtags, screen name, and the number of followers. We reject a tweet that is posted by a user with less than twenty followers. Filtering tweets is important to minimize spamming and trying to increase the quality of our corpus.

The next step is to stem the collected data before indexing them. In this step, we use the component developed by the first author to perform the stemming task. We also used Solr to create our index.

By the end of this phase, we have millions of Arabic tweets that are collected, cleaned, linguistically processed, and indexed.

We did not have the human resources to manually label the tweets (whether it is negative or positive). In our approach, we utilize the linguistic processing and Information retrieval methods to label our large size of training data set.

The process of automatically labeling our training data follows the following steps:

- Select a set of the most popular negative words (negative seed words)
- Select a set of the most popular positive words (positive seed words)
- Construct a negative search query using Solr search engine
- Construct a positive search query using Solr search engine
- Ranking the result using information retrieval approaches such as TF-IDF and cosine similarity.
- Select highest subset of the results
- Remove duplication
- Label the result.

To perform the above steps, we asked a group of 10 students to write down a set of popular negative and positive words. The most common ones are selected. For each category, we have at least 50 search terms.

The stemming was useful in expanding the search and increasing the recall since the results include all variations of the original search term. For example, the positive term افتخار (Honor):

Some variations of the term Honor (افتخار)
الإفتخار (The pride or honor)
وافتخر (And proud)
افتخارهم (Their pride)
وافتخاري (And my pride)
افتخرت (you were proud)
تفتخري (you are proud)
يفتخرون (They are proud)
فلنفتخر (Let us be proud)

We controlled the relevancy and accuracy of the retrieved tweets by selecting results that contain at least two positive terms. This measure increases our confidence in the retrieved tweets.

Additionally, we are utilizing one of the most essential features of IR systems which are controlling the relevancy to ensure that the "best" results are listed at the top. Relevancy control is achieved by computing of scores on both the query and the documents. Then a cosine similarity is found on the vector space model of both the query and document. Based on this, we only extract the results that appear early in the retrieved list of positive or negative tweets. So, we only select a subset of the result to increase the likelihood of its positiveness or negativeness.

From another perspective, this approach enriches our training data set with new terms that are not included the negative or positive query. For demonstration, suppose the negative seed contains the following two negative terms:

Negative Arabic term	English equivalent
حرامي	thief
نصاب	fraudulent

The top 10 tweets returned in response to the query containing only two negative terms is shown in table1.

Table 1. List of 10 negative tweets

Top Negative tweets	New negative terms
بلد المليون حرامي المليون نصاب والمليون مزوّر	مزوّر
حرامي نصاب الكلب خالد	الكلب
نصاب حرامي الكذاب واضح من كلامه سبحان الله	الكذاب
لعنة الله على كل نصاب وحرامي	لعنة الله
نصاب حرامي اهبل يدور الرتويت	اهبل
كذب في كذب المقاول نصاب والمتعهد حرامي وصاحب الشركه مجنون يكفي كذب	كذب, مجنون
فعلا أثبت لي انك "مخنث" و حرامي و نصاب و اقول ايه وانت فيك كل العبر.. أتفوووه	مخنث
اي مواطن ينتخب شخص نصاب فاشل حرامي منافق ——— اقوله انت اما مرتشي او مستفيد او غبي	فاشل، منافق، مرتشي، غبي
مجرم قتال قتلا حرامي نصاب بدون ذمه وشرف وبلا اخلاق حيوان حقير	مجرم، قتال، قتلا، بدون ذمه، وبلا اخلاق، حيوان، حقير
من الغباء ما قتل ، ومفيش حرامي او نصاب او مخنس . الا وبيترك ورائه بصمة حتي ولو بعد حين	الغباء، قتل، مخنس

We observe from the above table that the results contain 23 more negative expressions. Thus, this approach enriches the data set with more negative or positive terms without any human intervention.

5 Conducting the Experiment

To perform this experiment, we used Apache Mahout and Solr. The first phase is to collect and index tweets that are collected using Twitter API. The tweets were filtered to include only Arabic one. Using JAVA, we developed the software to clean, tokenize, stem, and index the tweets.

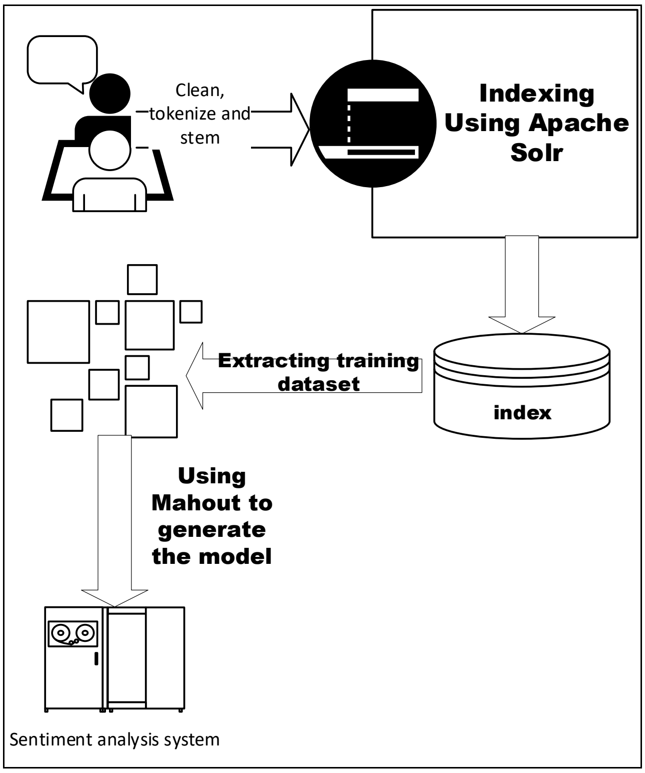

Fig. 3. The overall structure of the system

The system consists of two phases: the first one is collecting and annotation of positive and negative tweets. In this stage, we used Java and Solr to perform this Task. By following the technique described above, we extracted 100 thousand negative tweet and 100 thousands positive ones.

The next phase is training our system using the collected data set. We implement this phase using Mahout and Java.

6 Evaluation and Results

To evaluate our approach, we employ the accuracy, precision and recall measurements.

Accuracy

The percentage of correctly classified test samples

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP is True Positive, TN is True Negative, FP is False Positive, and TN is True Negative.

Precision

The fraction of true positives against all positives in classified results.

$$Precision\ = \frac{TP}{TP + FP}$$

Precision

The fraction of true positives against all positives in classified results.

$$Recall\ = \frac{TP}{TP + FN}$$

To conduct the evaluation, we have used two data sets, each one consisting of 2000 tweets that are manually annotated. The first was prepared by a Jordanian researchers [12]. We noticed that most of the tweets are being posted in Jordanian dialect. However, some are written in MSA. We refer to it by testJO set.

The results obtained from our system using testJO are:

Accuracy = 66.4 %

Precision = 67.8 %

Recall = 63.5 %

We also collected our own training set, which is also consisting of 2000 tweets (TestSA). It was labeled manually. This set is selected randomly, so it is not biased to any specific dialect. However, it is well known that the majority of Arabic tweets is written in Gulf dialects, especially the Saudi one.

The results obtained from our system using testSA are:

Accuracy = 97.7%

Precision = 98.3 %

Recall = 97.1 %

We observe that our system perform very well with testSA. This is because the training data set is extracted from twitter without any preference to any sublanguage or dialect. However, it seems that the distance in terms of structure and lexicon is significant between Jordanian dialect and the dominated ones in Twitter.

7 Evaluation and Results

In this paper, we discussed techniques of labeling a large amount of training data set without human intervention. This is essential in building and enhancing Arabic sentiment research. We use machine learning approach using Naive Bayes. We enhance this model by using TF-IDF scheme. The evaluation of these presented techniques and approaches shows that it is possible to achieve very high performance. However, the training data and the testing one should be linguistically close.

8 References

[1] D. Daoud, A. Al-Kouz, And M. Daoud, "Time-Sensitive Arabic Multiword Expressions Extraction From Social Networks," International Journal Of Speech Technology (Ijst), 2015. https://doi.org/10.1007/s10772-015-9315-3

[2] D. Daoud, A. Al-Kouz, K. Hasssan, And L. Milliam, "Arabic Tweets Clustering And Labeling Based On Lingual And Semantically Enriched Bayesian Network Model," Recent Patents On Computer Science, Vol. 8, Pp. 1-14, 2015. https://doi.org/10.2174/2213275908999150324162827

[3] A. Hamdi, K. B. Shaban, And A. Zainal, "A Review On Challenging Issues In Arabic Sentiment Analysis," Jcs, Vol. 12, Pp. 471-481. https://doi.org/10.3844/jcssp.2016.471.481

[4] A. Shoukry And A. Rafea, "Sentence-Level Arabic Sentiment Analysis," Presented At Collaboration Technologies And Systems (Cts), 2012 International Conference On. https://doi.org/10.1109/CTS.2012.6261103

[5] S. Rosenthal, N. Farra, And P. Nakov, "Semeval-2017 Task 4: Sentiment Analysis In Twitter," Presented At Proceedings Of The 11th International Workshop On Semantic Evaluation (Semeval-2017). https://doi.org/10.18653/v1/S17-2088

[6] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, And M. Al-Ayyoub, "Arabic Sentiment Analysis: Lexicon-Based And Corpus-Based," Presented At Applied Electrical Engineering And Computing Technologies (Aeect), 2013 Ieee Jordan Conference On. https://doi.org/10.1109/AEECT.2013.6716448

[7] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, M. Al-Ayyoub, M. N. Al-Kabi, And S. Al-Rifai, "Towards Improving The Lexicon-Based Approach For Arabic Sentiment Analysis," International Journal Of Information Technology And Web Engineering (Ijitwe), Vol. 9, Pp. 55-71. https://doi.org/10.4018/ijitwe.2014070104

[8] S. Ahmed, M. Pasquier, And G. Qadah, "Key Issues In Conducting Sentiment Analysis On Arabic Social Media Text," Presented At Innovations In Information Technology (Iit), 2013 9th International Conference On. https://doi.org/10.1109/Innovations.2013.6544396

[9] M. Al-Ayyoub, S. B. Essa, And I. Alsmadi, "Lexicon-Based Sentiment Analysis Of Arabic Tweets," International Journal Of Social Network Mining, Vol. 2, Pp. 101-114. https://doi.org/10.1504/IJSNM.2015.072280

[10] L. Albraheem And H. S. Al-Khalifa, "Exploring The Problems Of Sentiment Analysis In Informal Arabic," Presented At Proceedings Of The 14th International Conference On Information Integration And Web-Based Applications & Services.

[11] S. R. El-Beltagy And A. Ali, "Open Issues In The Sentiment Analysis Of Arabic Social Media: A Case Study," Presented At Innovations In Information Technology (Iit), 2013 9th International Conference On. https://doi.org/10.1109/Innovations.2013.6544421

[12] A. N. A., M. N. A., S. M., Al-Ayyoub M., And "Arabic Sentiment Analysis: Corpus-Based And Lexicon-Based," Presented At Ieee Conference On Applied Electrical Engineering And Computing Technologies (Aeect 2013), Amman, Jordan., 2013.

9 Authors

Daoud M. Daoud received his B.Sc. degree in Electrical and Computer Engineering from Kuwait University in 1988, his M.Sc. in Computing Science from Glasgow University, UK, and his PhD in Computing Science from Joseph Fourier University, France. Dr Daoud is currently serving as an Associate Professor at PSUT. He is also a faculty member at Higher colleges for technology, UAE. In the period between 1996 to1999, Dr Daoud worked as a principal investigator for the Arabic section of the Universal Networking Language project. He then worked at the Institute of Advanced Studies, United Nations University (1998-1999). He served as a director for Next Generation, in the Services department at Paltel (1999-2001). His main research interests are Natural Language Processing, machine translation, Information Extraction, Information Retrieval and analysis of Arabic Social Media and Big Data. Email: ddaoud@hct.ac.ae

Professor M Samir Abou El-Seoud received his BSc degree in Physics, Electronics and Mathematics from Cairo University in 1967, his Higher Diplom in Computing from Technical University of Darmstadt (TUD) /Germany in 1975 and his Doctor of Science from the same University (TUD) in 1979.
Field of study: Scientific Computations and Parallel Algorithms.
Research interests: Computer Aided Learning, Parallel Algorithms, Numerical Scientific Computations and Computational Fluid Mechanics

Professor El-Seoud held different academic positions at TUD Germany. Letest Full-Professor in 1987. Outside Germany, Professor El-Seoud spent different years as a Full-Professor of Computer Science at SQU – Oman, Qatar University, and PSUT-Jordan and acted as a Head of Computer Science for many years. At industrial institutions, Professor El-Seoud worked as Scientific Advisor and Consultant for the GTZ in Germany and was responsible for establishing a postgraduate program leading to M.Sc. degree in Computations at Colombo University / Sri-Lanka (2001 – 2003). He also worked as Application Consultant at Automatic Data Processing Inc., Division Network Services in Frankfurt/Germany (1979 – 1980). Professor El-Seoud joined The British University in Egypt (BUE) in 2012. Currently, he is Basic Science Coordinator at the Faculty of Informatics and Computer Science (ICS) at BUE. Professor El-Seoud has more than 150 publications in international proceedings and reputable international journals. Email: samir.elseoud@bue.edu.eg, saoudi@hct.ac.ae

Article submitted 2020-02-06. Resubmitted 2020-03-17. Final acceptance 2020-03-22. Final version published as submitted by the authors.