I.J. Information Engineering and Electronic Business, 2019, 6, 16-23
Published Online November 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2019.06.03

A Corpus Based Approach to Build Arabic Sentiment Lexicon

Afnan Atiah Alsolamy[1], Muazzam Ahmed Siddiqui[2], Imtiaz Hussain Khan[3]
[1,2]Department of Information Systems, King Abdulaziz University, Saudi Arabia
[3]Department of Computer Science, King Abdulaziz University, Saudi Arabia

Received: 17 September 2019; Accepted: 10 October 2019; Published: 08 November 2019

***Abstract*—**Sentiment analysis is an application of artificial intelligence that determines the sentiment associated sentiment with a piece of text. It provides an easy alternative to a brand or company to receive customers' opinions about its products through user generated contents such as social media posts. Training a machine learning model for sentiment analysis requires the availability of resources such as labeled corpora and sentiment lexicons. While such resources are easily available for English, it is hard to find them for other languages such as Arabic. The aim of this research is to build an Arabic sentiment lexicon using a corpus-based approach. Sentiment scores were propagated from a small, manually labeled, seed list to other terms in a term co-occurrence graph. To achieve this, we proposed a graph propagation algorithm and compared different similarity measures. The lexicon was evaluated using a manually annotated list of terms. The use of similarity measures depends on the fact that the words that are appearing in the same context will have similar polarity. The main contribution of the work comes from the empirical evaluation of different similarity to assign the best sentiment scores to terms in the co-occurrence graph.

***Index Terms*—**Sentiment analysis, Arabic sentiment lexicon, Propagation method, Similarity measure

I. Introduction

Nowadays, social media are changing how people share approximately everything in their life. It also extends to sharing their experiences of products they buy, service they ask, place they visit, movie they watch, and book they read. Many websites and social media are created only for sharing experiences and ideas with other users. This massive amount of opinions has to be analyzed to extract useful information and hidden knowledge from them. Sentiment analysis is an automatic process to extract and determine the polarity or sentiment of a sentence [1, 2, 3]. Sentiment is an expression of person feeling, view, and opinion toward an event or situation. The sentiment of the text could be classified in exact categories like positive, negative, or scale from strongly positive to strongly negative. Solving sentiment analysis needs various resources and tools. In Arabic language, sentiment analysis faces many issues and challenges [4, 5]. One of the main shortages is a lack of high-quality sentiment lexicon [4]. Sentiment lexicon is an instrumental and a contributory resource in the process of classifying text to either positive or negative. Sentiment lexicon contains terms with their polarity. The polarity comes in different forms, either positive or negative, or score indicating the strength of the terms. Another resource can be used in sentiment analysis, or other computational linguistics processes is a corpus. It is a collection of text structured in a way that helps in automatic processing.

We used two underlying principles to build the sentiment lexicon:

1. Words found in the same context are semantically similar to each other.
2. Similar words carry the same sentiment polarity.

The first of the above principles implies the need to identify similar terms. To achieve this, different similarity measures can be used. Similarity measures provide a numerical measure of the similarity or dissimilarity (distance) between two objects. The second principle implies that the sentiment polarity from a known list of words can be propagated to other words by exploiting the similarity relationship. This can be done by creating a term co-occurrence graph and using propagation algorithms to spread the similarity from known seed list of terms to other terms. The objectives of this research were to provide an empirical comparison of different similarity measures and devise a graph propagation algorithm that utilizes these measures to assign scores to terms in a co-occurrence graph. To create the co-occurrence graph a corpus based approach was chosen where terms found in a document are considered to be co-occurring terms. The main contribution of our work is that it compares different similarity measures to identify terms in Arabic language that should receive same polarity and validate the resulting lexicon against a manually labeled list.

II. Related Work

This work aims to build an Arabic sentiment lexicon. According to [6], there are three main approaches to construct an opinion lexicon, which are corpus-based,

dictionary-based, and manual approaches. In addition, some of the Arabic studies have utilized the availability of English sentiment lexicon, and they used translation API service to get Arabic word translated with its polarity [7, 8]. Based on the lexical resources, a dictionary-based approach builds the polarity lexicon. In [9], the researchers used a semi-supervised approach that depends on the idea that if terms in an online dictionary have similar gloss, then there is a high probability of them having the same polarity. A classifier is trained based on the small seed list, to categorize unassigned words as positive or negative class. In [10], the authors used a WordNet thesaurus to assign scores to lexicon words, reversing their intensity by measuring the strength of their graphical connections to the seed words. In another work [11], the authors built an Arabic sentiment lexicon based on Arabic WordNet resource. The building mechanism started with a small seed words' list followed by the propagation of sentiment score to words based on the utilization of the synsets of Arabic WordNet. The result of evaluation of the lexicon is promising with accuracy of 96%.

The corpus-based approach depends on the collection of text and tries to predict the sentiments of words within it to build the lexicon. Many methods can be used, such as looking for conjugated adjectives within English text [12] and Arabic text [5]. The adjectives conjoined with "AND" have the same polarity, whereas the adjectives conjoined with "BUT" have different polarity values. Another method based on the previous idea is presented in [13]. It extends the idea by looking not only into intra-sentence, but also in inter-sentence. In other words, neighboring sentences (before and after) are looked at to expand the sentiment word list. Similarity graph and label prorogation methods can also be used to build the sentiment lexicon for English [14] and Arabic [15]. Another method depends on statistical techniques to build the lexicon by measuring the co-occurrence and similarity of the words. Pointwise mutual information (PMI) and Chi-square tests have been used in [16] and, PMI in [17] to measure the polarity of the lexicon terms. According to [18], the sentiment lexicon can be categorized into two different types: a general-purpose sentiment lexicon and a context-dependent sentiment lexicon. The general purpose has only a sentiment word with assigned score that represents the degree of polarity [18]. Whereas the context-dependent lexicon includes the sentiment word attached to different aspects of the studied domain [18]. The polarity of the word could be different according to the attached aspect, such as (long boot) is negative and (long battery) is positive. The context-dependent sentiment lexicon can handle the issue of the same words that have different polarities based on the domain and the context [19]. In [19], the authors have studied both aspect and opinion words to build the context-dependent sentiment lexicon. In addition, they add some linguistic patterns to handle the changing of the polarities. The work in [18] assigns polarities to the entries of context-dependent sentiment lexicon based on an objective function with defined constrains: sentiment score, synonyms and antonyms, general-purpose sentiment lexicon, and linguistics patterns.

Recently, Arabic sentiment analysis and sentiment lexicon construction has received significant attention. In an interesting work [20], the researcher presented a large-scale sentiment lexicon called SLSA. They constructed their dataset by linking AraMorph lexicon with SentiWordNet. They evaluated the lexicon and reported high accuracy results. In [21], a weighted lexicon-based algorithm is proposed to build an Arabic lexicon that exploits association rules to assign weights to the words in the lexicon. In [22], the authors built a large-scale Arabic sentiment lexicon called AraSenTi. The lexicon consists of Twitter data, which is drawn from various Saudi and other Arabic dialects. In another study [23], the authors presented a sentiment lexicon consisting of around 6000 tweets in modern standard Arabic. They used trending hash tags in Saudi Arabia to collect their data. The tweets are labeled as positive, negative or neutral. In [24], the authors used two Arabic corpora consisting of various reviews to build a sentiment lexicon. They developed a sentiment lexicon comprising ca. 9000 Arabic reviews obtained from 15 different domains. The data were annotated by two native speakers, yielding around 5400 positive reviews and 3450 negative reviews. In [25], the authors used semi-supervised learning approach to build a sentiment lexicon for Arabic language. Their primary source of data was Arabic WordNet. In [26], the researchers presented a large-scale sentiment and emotion lexicon consisting of around 32000 words in Arabic. Their lexicon is built on an existing Arabic sentiment lexicon called ArSenL. They evaluated their lexicon on SeEval 2018 shared task and report on encouraging accuracy results.

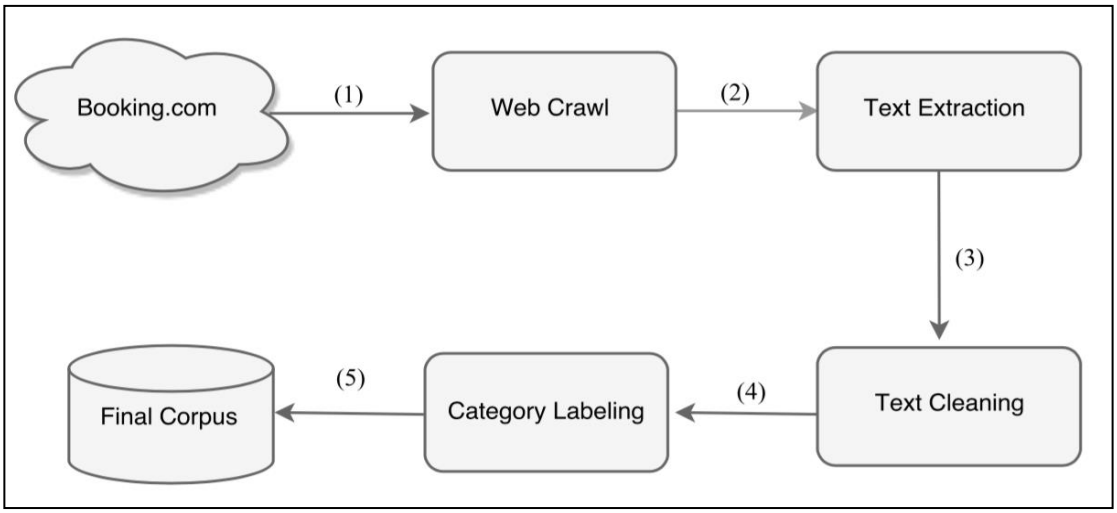

Fig.1. Corpus Building Steps

III. Building the Corpus

A sentiment lexicon can be built by manually annotating terms with their sentiment, but the approach clearly suffers with scalability issues. Thus, an automated method using a corpus or dictionary is preferable. Between these two approaches, the latter one relies on an existing dictionary and the scoring is limited to the terms occurring in the dictionary. This limits the size of the lexicon to the size of the dictionary used. The corpus-based approach relies on an unannotated corpus thus providing a clear advantage over the dictionary-based approach. Therefore, the first task of this work was to build the corpus. which involves various steps as depicted in Fig. 1.

3.1 Web Crawling

To build any corpus, there is a need of huge amount of text from the Web. Collecting this huge amount of text is a cumbersome task and hard to be done manually. In order to do that, there is a need of automatic mechanism that do traversing like a spider and traverse between web pages and download them for different purpose. This mechanism is called Web crawling. Here, we wrote a small script to follow links of booking.com website to reach customers reviews web pages and then download them.

3.2 Text Extraction

The downloaded information from webpages is written using hypertext markup language (aka HTML). Extracting this information from the Internet manually is not an easy task. There is a need to harvest the information from unstructured format into structured format that can be easily used and analyzed which is the second step. Numerous reviews from 692 hotels have been used. Each hotel reviews have been collected in a single web page. So, there are separate 692 web pages. The first step was to combine all these different reviews into one single file to ease the manipulation and extracting the exact information. This is done using small script written in Python. Moreover, the combined file contains 232,284 different reviews or documents from 692 hotels.

3.3 Text Cleaning

The third step is text cleaning which involves various activities. One of them is deleting empty reviews. Actually, there are some reviewers who just rate the hotel without mentioning anything either positive comment or even negative. Therefore, there is necessary to delete this rating before further processing. After deleting these empty reviews, the number of documents now is 165,994. Although the collected booking customer reviews are from Arabic version, some of reviews are written in various languages including English, Japanese etc. In addition, the reviews contain non-language tokens like emoticons that need to be excluded from the corpus. For achieving this, predefined function in natural language tool kit in Python has been used. After the necessary cleaning, we have finally 132,231 documents.

3.4 Category Labeling

Actually, booking.com uses twelve Likert scales, which the reviewer can assign to their review. The Likert scales are listed in the first column in Table 1. In addition, it divides two sections for each review, one for the positive side and the other for the negative aspect. For saving a corpus into text files, we created a folder for each Likert scale. In addition, we created two sentiment sub-folder (positive, negative) under each Likert scale folder. After that, we save each review as a text file under their corresponding folders. Table 2 presents a distribution of positive and negative reviews to the eleven categories.

Table 1. A Distribution of Positive and Negative Reviews to the Booking Categories (The data source is booking.com)

English Likert Scale	Arabic Likert Scale	Number of Reviews	Number of Positive Reviews	Number of Negative Reviews
Exceptional	استثنائي	27,362	16,033	11,329
Excellent	ممتاز	9,187	4,684	4,503
Wonderful	رائع	12,220	6,493	5,727
Very good	جيد جدا	9,788	4,884	4,904
Good	جيدا	25,571	12,404	13,167
Pleasant	مرضي	13,618	6,253	7,365
Fair	حسن	10,720	4,753	5,967
Okay	مقبول	5,659	2,579	3,080
Disappointing	مخيب للآمل	8,322	3,579	4,743
Poor	ضعيف	5,899	2,475	3,424
Very poor	ضعيف جدا	3,867	1,547	2,320
Total		**132,231**	**65,684**	**66,529**

Table 2. Corpus Statistic ((The documents in our corpus are drawn from booking.com)

Measurement	Number
Number of documents	132,231
Number of all tokens	1,332,409
Number of unique tokens	13,493

The text preprocessing for Arabic language is a complex task due to various reasons. For example, Arabic language has many inflectional morphology systems. Second, Arabic spelling involves somewhat complicated phenomenon because different rules and standard affect the spelling of words. We used a special tool for Arabic natural language processing (NLP) called MADAMIRA for Arabic text preprocessing task. Arabic language has three different classifications, Modern Standard Arabic (MSA), Classical Arabic, and Dialectal Arabic. MSA is the language of news and formal writing and education. Whereas, Dialectal Arabic is the main language that is used in social media and informal conversation. In fact, the Dialectal Arabic is different than MSA in various forms including morphology and inflectional system. In addition, the number of modern dialects is big also. These reasons decrease the accuracy of any NLP tools for Arabic language [27]. MADAMIRA is the incorporation and improvement of two worthy systems in Arabic NLP called MADA and AMIRA. MADAMIRA provides great NLP functions that help in disambiguation of word level of MSA text as well as Egyptian dialect. MADAMIRA extracts the tokens according to different scheme and rules. In addition, it defines the lemma and stem forms of a token. In addition, it defines part of speech tagging and full set of morphological features.

As mentioned earlier, the corpus has 132,231 different documents. MADAMIRA supports both XML and plain text file as an input for the tool. A small Python script has been written to read corpus text files one by one and save them in one consolidate file in MADAMIRA XML format. After running MADAMIRA, there are 1,332,409 tokens with unique 13,493 words. This information is presented in Table 2.

IV. Building the Sentiment Lexicon

Building the sentiment lexicon can be achieved using different techniques and methods. As mentioned earlier, there are three approaches to build the sentiment lexicon: dictionary based, corpus based and manual. To build the sentiment lexicon using corpus-based approach, we need certain preconditions. First and foremost, we need to

have an Arabic corpus. Once the corpus is available, in the next step, we need to build the sentiment lexicon. Fig. 2 illustrates the main steps of building sentiment lexicon.

In this work, vector space model (VSM) has been employed to represent the words in corpus. Each document in VSM is represented as a collection of points in a space. Logically, the points that are closely related have similar characteristics. Whereas, the points that are far away from each other are dissimilar. According to [28], VSM is an easiest way to extract knowledge from the corpus. In addition, it needs less work comparing to other semantic approaches. This research uses the concept of VSM and builds the word-word matrix. A word-word matrix is a mathematical representation of frequency of the word in the corpus. The word-word matrix is an n x n matrix, where each row and column represents one word. The value of any cell (i, j) is the count of the number of times both words occur together in the same documents. After building a word-word matrix, there is a need to measure the similarity between each vector with all the remaining vectors. The vectors that have high similarity value indicate that the corresponding words are also very similar. Similarity, methods can help to build the sentiment lexicon by categorizing the words' polarities and calculating the degree of similarity of words. Three similarities methods have been used. Cosine similarity, Jaccard similarity and Dice similarity. Cosine similarity metric is the most popular method to measure the similarity in the text documents, information retrieval and clustering [29]. The cosine similarity measures the normalized dot product of two vectors. It determines the cosine value of the angle between two vectors. The cosine value 0 means that the angle is 90 °, therefore, the two documents do not share any common characteristics. Whereas, cosine value 1 means that the two vectors are identical and have the same characteristics. The Jaccard similarity is a statistical method used for comparing the similarity and dissimlarity of characteristics between samples of information sets. It is defined as the value of the intersection or sharing number of characteristics between the sets under study divided by the number of all characteristics (union of the sample set) [30, 31]. The Dice similarity is the total summation of dot product of two vectors multiplied by two.

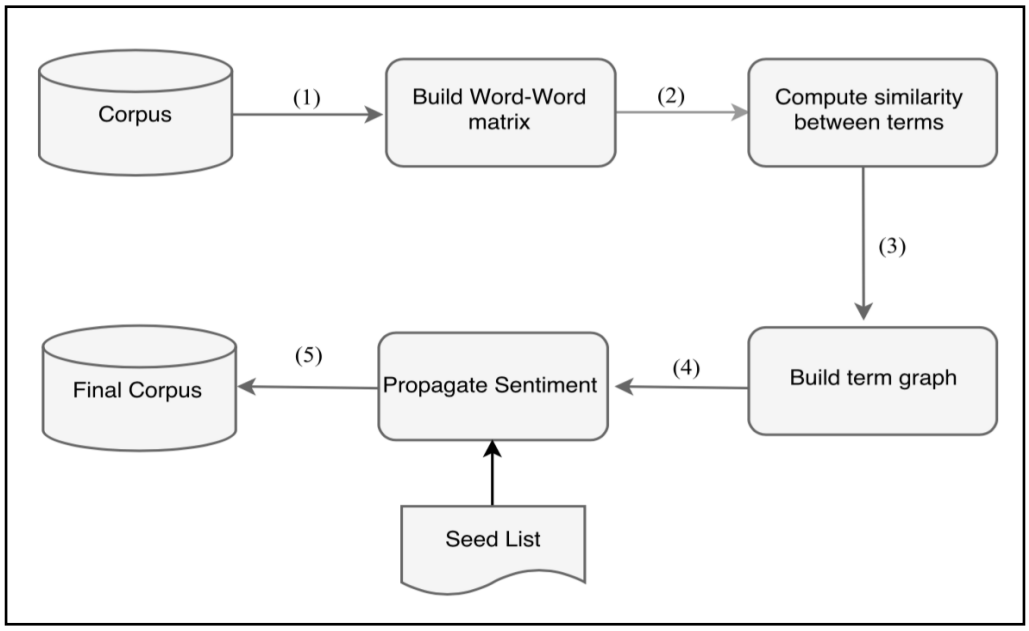

Fig.2. Building Sentiment Lexicon Steps

Algorithm1: Propagation Algorithm

Input:

G(E, V) → *# Weighted graph where w_{ij} is the weight of edge (v_i, v_j); G encode similarities between its nodes.*

P →*# Positive seed set*

N→ *# Negative seed set*

T→ *# Number of iterations*

Output:

POL → *#Final sentiment lexicon*

Initialize:

$\alpha_{ii} = 1$, $\alpha_{ij} = 0$, F = {}, i= j = k =0, Pol- = 0, Pol+ = 0

Body:

1. Set $\alpha_{ii} = 1$, and $\alpha_{ij} = 0$ for all $i \neq j$ *# Set 1 for same nodes and zero for different nodes*
2. For $v_i \in P$: *# for every node in Positive/ Negative seed list*
3. $F=\{v_i\}$ *# Assign positive/negative node to F*
4. For $t \in T$: *# for every iteration defined*
5. For $(v_k, v_j) \in E$ where $v_k \in F$ *# for every two vectors in graph (members of E)*
6. $\alpha_{ij} = \max \{\alpha_{ij}, \alpha_{ik}. w_{kj}\}$ *# Calculate the max weighted paths from seed to vector.*
7. $F = F \cup \{v_j\}$ *# Add this vector to F and go to next vector*
8. For $v_j \in V$: *# for every vector in Graph*
9. $Pol+[j] = \sum \alpha ij$ *# Calculate vector polarity as summation of all its best paths*
10. Repeat steps 1-9 to compute Pol- *# Repeat the same steps to calculate the negative polarity*
11. POL= [Pol+]-[Pol-] *# The final Polarity is the difference between positive and negative*

Table 3. Number of Different Sentiment Lexicon Entries

	Number of Positive Terms	Number of Negative Terms	Number of Neutral Terms
Cosine Similarity	6435	6673	385
Jaccard Similarity	6466	7004	25
Dice Similarity	6589	6966	0

Table 4. Recall, Precision and F-measure for Different Similarities

	Recall for Positive Terms	Recall for Negative Terms	Precision for Positive Terms	Precision for Negative Terms	F-measure for Positive Terms	F-measure for Negative Terms
Cosine Similarity	97%	61%	56%	99%	70%	75%
Jaccard Similarity	60%	60%	66%	66%	63%	63%
Dice Similarity	57%	58%	67%	61%	62%	59%

The last step to build the lexicon is to use propagation technique of small seed list of positive and negative words. Our approach is very similar to the one discussed in [32]. The work in [32] investigated the visibility of building sentiment lexicon from unlabeled web documents based on graph built from co-occurrence statistics from the Web. Then it implements the graph propagation algorithm over that.

The input to the graph propagation used in this work is a generated similarity matrix, positive seed list, negative seed list and the number of iterations to perform. The pseudo-code of our propagation method is listed in Algorithm 1.The algorithm computes a positive and a negative polarity magnitude for each word in the graph. It is equal to the sum over the max weighted path from every seed word to word. The graph propagation algorithm calculates the sentiment of each word as the aggregate of all the best paths to seed words. After running the above, there are two polarity magnitudes for each word, one positive and one negative. Therefore, the final polarity is the difference between both; the higher one will get the score. The output of Algorithm 1 is sentiment lexicon for three similarity measurements as shown in Table 3.

V. Lexicon Evaluation

A sample of 2500 words has been labeled by an Arabic native speaker is used as a base for testing the generated lexicon. This sample consists of 1121 positive words, 1124 negative words and 255 neutral words. We test the generated lexicon against the labeled sample. Along with testing using labeled annotated sample, precision and

recall are important performance measures in document retrieval. Precision measures how many returned documents are correct (Equation 1). In sentiment lexicon case, it measures how many labeled words matched with the lexicon polarity. Whereas, recall measures the number of correct documents that have been returned (Equation 2). In sentiment lexicon case, it measures how many correct labeled words have been returned. They show numbers of predicted positive and negative words (annotated sample) along with the number labeled words of the lexicon.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

Another important performance measurement is f-measure. F-measure is the harmonic average of the precision and recall as computed in Equation (3). It is a trade-off between precision and recall values. The f-measure for all three similarities used in this research (cosine, Jaccard and dice) is around their recall and precision values.

$$\text{F-measure} = 2 \times \frac{Precision \times Recall}{Precison + Recall} \quad (3)$$

The results of performance measure are shown in Table 4. In cosine similarity, the recall for positive entries is very high but the precision is very low. Whereas, the precision is very high for negative but recall is very low. In Jaccard similarity, the values of precision and recall are very close. In dice, they are also very close to each other. The cosine has higher f-measure than the other similarities with 70% for positive and 75% for negatives entries. In Jaccard similarity, the value is matching for both entries. Whereas, f-measure for dice positive words is higher than negative ones. The results in Table 4 elucidate that cosine similarity is a better similarity measure than dice and Jaccard similarity measures. It is clear from the results in Table 4 that precision and recall are not reasonably high. One interpretation of these low scores is that Arabic language has rich morphology and poses significant challenges on the language processing tasks, including sentiment analysis.

VI. Conclusion

This paper proposed an algorithm to build an Arabic sentiment lexicon by exploiting semantic similarity between co-occurring terms in a corpus. The similarity can be computed using different measures and the aim was to compare the most commonly used similarity measures that result in best sentiment score assignments to terms. Our results indicated that cosine similarity outperformed dice and Jaccard similarity measures as it incorporated more information by using term frequencies instead of binary occurrence only.

This work can be extended into many directions, including for example, expanding the lexicon with Arabic slang and dialect and evaluation of the generated Arabic lexicon in sentiment classification

References

[1] M. Korayem, D. Crandall and M. Abdul-Mageed, "Subjectivity and sentiment analysis of arabic: A survey," in *Advanced Machine Learning Technologies and Applications*, 2012.

[2] L. Dey, S. Chakraborty, A. Biswas, B. Bose and S. Tiwari, "Sentiment Analysis of Review Datasets Using Naïve Bayes and K-NN Classifier.," *International Journal of Information Engineering and Electronic Business.*, vol. 4, pp. 54-62, 2016.

[3] S. O. Oppong, D. Asamoah, E. O. Oppong and D. Lamptey, "Business Decision Support System based on Sentiment Analysis.," *International Journal of Information Engineering and Electronic Business.*, vol. 1, pp. 36-49, 2019.

[4] Shoukry and A. Rafea, "Sentence-level Arabic sentiment analysis," in *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, 2012.

[5] S. R. El-Beltagy and A. Ali, "Open Issues in the Sentiment Analysis of Arabic Social Media: A Case Study," in *Innovations in Information Technology (IIT), 2013 9th International Conference on*, 2013.

[6] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1-167, 2012.

[7] M. Abdul-Mageed and M. T. Diab , "Toward building a large-scale Arabic sentiment lexicon," in *Proceedings of the 6th International Global WordNet Conference*, 2012.

[8] N. A. Abdulla, N. A. Ahmed, M. A. Shehab and M. Al-Ayyoub, "Arabic Sentiment Analysis: Lexicon-based and Corpus-based," in *Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on*, 2013 .

[9] Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss classification," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005.

[10] S. Blair-goldensohn, T. Neylon, K. Hannan, G. A. Reis, R. Mcdonald and J. Reynar, "Building a sentiment summarizer for local service reviews," in *WWW Workshop on NLP in the Information Explosion Era*, 2008.

[11] F. Mahyoub , M. Siddiqui and M. Dahab , "Building an Arabic Sentiment Lexicon Using Semi-supervised Learning," *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 4, pp. 417-424, 2014.

[12] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 1997.

[13] H. Kanayama and T. Nasukawa , "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006.

[14] L. Velikovich, S. Blair-Goldensohn , K. Hannan and R. McDonald, "The viability of web-derived polarity lexicons," in *Human Language Technologies: The 2010*

Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010.
[15] M. Elhawary and M. Elfeky, "Mining arabic business reviews," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, 2010.
[16] N. Kaji and M. Kitsuregawa, "Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents," in *Conference on Empirical Methods in Natural Language Processing*, 2007.
[17] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002.
[18] Y. Lu, M. Castellanos, U. Dayal and C. Zhai, "Automatic construction of a context-aware sentiment lexicon: an optimization approach," in *Proceedings of the 20th international conference on World wide web*, 2011.
[19] X. Ding , B. Liu and P. S. Yu , "A holistic lexicon-based approach to opinion mining," in *WSDM '08 Proceedings of the 2008 International Conference on Web Search and Data Mining*, 2008.
[20] R. Eskander and O. Rambow, "SLSA: A Sentiment Lexicon for Standard Arabic.," in *Proceedings of the 2015 conference on empirical methods in natural language processing.*, Lisbon, Portugal, 2015.
[21] Assiri, A. Emam and H. Al-Dossari, "Towards enhancement of a lexicon-based approach for Saudi dialect sentiment analysis.," *Journal of information science.*, vol. 44, no. 2, pp. 184-202, 2017.
[22] A.-T. Nora, H. Al-Khalifa and A. AlSalman, "AraSenTi: Large-scale Twitter-specific Arabic sentiment lexicons.," in *Proceedings of the 54th annual meeting of the association for computational linguistics.*, Berlin, Germany, 2016.
[23] Al-Thubaity, Q. Alqahtani and A. Aljandal, "Sentiment lexicon for sentiment analysis of Saudi dialect tweets.," in *Proceedings of the 4th international conference on Arabic computational linguistics.*, Dubai, United Arab Emirates, 2018.
[24] T. Al-Moslmi, M. Albared, A. Al-Shabi, N. Omar and S. Abdullah, "Arabic senti-lexicon: Constructing publicly available language resources for Arabic sentiment analysis.," *Journal of information science.*, vol. 44, no. 3, pp. 345-362, 2017.
[25] K. Sabra, R. Zantout, M. El Abed and L. Hamandi, "Sentiment analysis: Arabic sentiment lexicons.," in *Proceedings of sensors networks smart and emerging technologies.*, 2017.
[26] G. Badaro, H. Jundi, H. Hajj, W. El-Hajj and N. Habash, "ArSEL: A large scale Arabic sentiment and emotion lexicon.," in *Proceeding of the 3rd workshop on open-source Arabic corpora and processing tools.* , Proceeding of the 3rd workshop on open-source Arabic corpora and processing tools. , 2018.
[27] Pasha, M. Al-Badrashiny, M. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow and R. Roth, "MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic," in *Proceedings of the 9th International Conference on Language Resources and Evaluation*, 2014.
[28] P. Turney and P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics," *Journal of Artificial Intelligence Research*, no. 37, pp. 41-188, 2010.
[29] Huang, "Similarity Measures for Text Document Clustering," in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, 2008.
[30] M. Jayaram, P. G.K. and D. K.M, "Clustering of Ears based on Similarity Metrics for Personal Identification," *International Journal of Applied Engineering Research*, vol. 10, no. 12, pp. 30927-30942, 2015.
[31] J. Singthongchai and S. Niwattanakul, "A Method for Measuring Keywords Similarity by Applying Jaccard's, N-Gram and Vector Space," *Lecture Notes on Information Theory*, vol. 1, no. 4, December 2013.
[32] L. Velikovich, S. Blair-Goldensohn, K. Hannan and R. McDonald, "The viability of web-derived polarity lexicons," in *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

Authors' Profiles

Afnan Atiah Alsolamy is a Master's student in the Department of Computer Science at King Abdulaziz University. Her research interests are in sentiment analysis.

Dr. Imtiaz Hussain Khan is an Associate Professor in the Department of Computer Science at King Abdulaziz University, Jeddah, Saudi Arabia. He received his master's degree in computer science from the University of Essex, UK, in 2005. He earned his Ph.D. in artificial intelligence from the University of Aberdeen, UK, in 2010. His research interests are natural language processing and sentiment analysis.

Dr. Muazzam Ahmed Siddiqui is an associate professor at the Faculty of Computing and Information Technology, King Abdulaziz University. He received his BE in electrical engineering from NED University of Engineering and Technology, Pakistan, and MS in computer science and PhD in modeling and simulation from University of Central Florida. His research interests include text mining, information extraction, data mining and machine learning.

How to cite this paper: Afnan Atiah Alsolamy, Muazzam Ahmed Siddiqui, Imtiaz Hussain Khan, " A Corpus Based Approach to Build Arabic Sentiment Lexicon", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.11, No.6, pp. 16-23, 2019. DOI: 10.5815/ijieeb.2019.06.03

