The Effect of using Light Stemming for Arabic Text Classification

Jaffar Atwan[1]
Department of Computer Information Systems
Al-Balqa Applied University
Al-Salt, Jordan

Mohammad Wedyan[2]
Faculty of Artificial Intelligence
Al-Balqa Applied University
Al-Salt, Jordan

Qusay Bsoul[3]
Faculty of Science and Technology
Universiti Sains Islam Malaysia
Bandar Baru Nilai, Malaysia

Ahmad Hamadeen[4]
Department of Computer Science
Al-Balqa Applied University
Al-Salt, Jordan

Ryan Alturki[5]
Department of Information Science
College of Computer and Information Systems
Umm Al-Qura University, Makkah, Saudi Arabia

Mohammed Ikram[6]
Computer Science Department
University College in Al-Jamoum
Umm Al-Qura University

***Abstract*—Arabic is one of the Semitic languages in antiquity and one of the six official languages of the UN. Also, Arabic classification plays a significant and essential role in modern applications. There is a big difference between handling English text and Arabic text classification; preprocessing is also challenging for Arabic text. This paper presents the implementation of a Naïve Bayes classifier for Arabic text with and without stemmer. A set of four categories and 800 documents were used from the Text Retrieval Conference (TREC) 2001 dataset. The results showed that Naïve Bayes with light stemmer achieves better results than Naïve Bayes without stemmer. The findings of the classifier accuracy by employing stemmer and without stemmer are as preprocessing. It reveals that the accuracy resulted from the light stemmer was better than the classifier without stemmer detection, which Naïve Bayes Classification with light stemmer got 35.0745 higher than the Naïve Bayes Classification 33.831% without stemmer. After contrasting them, the stemmer got better accuracy than the classifier.**

Keywords*—Arabic language; light stemming; information retrieval; Naïve Bayes classification

I. Introduction

Machine-readable information is available in large and increasing quantities that complicate comprehension and use. Machine learning (ML) provides tools that help organize vast numbers of texts and detect them automatically [1]. Feature selection is the most significant application in ML. Feature selection, by identifying the most salient features for learning, the interest of the learning algorithm sheds the lights on the most valuable data for analysis and future prediction [1-4]. The ideas were adopted from test theory to develop a technique for correlation-based feature selection and estimate a group of ML algorithms that taught a different group of natural and artificial problems. The feature selection is easy and fast to implement; it removes irrelevant and repetitive data and its ability to enhance the learning algorithms' performance in many cases. The results of this technology can be contrasted with a modern feature selector elicited from the literature but require much less computation. In the current research, the domain of Arabic documents will be used to decrease features number and promote the performance detection of the Arabic document.

There is no comprehensive definition of the Arabic language. Some Arabic language documents or emails may have some unwanted words or samples; this email detects spam due to unwanted email messages; not all Arabic emails are spam. And the commercial email that is spam [5], spam as the name suggests, is unwanted emails, but there is a general question, what are junk emails? Although Arabic is known to many users, it is difficult to see the definition of both the Arabic language and unwanted messages.

This paper is organized as the following. Section 2 provides a review of previous related studies. The background of the study on Arabic classifiers is tackled by Section 3. The steps followed during the experiment present in Section 4. Section 5 shows the experimental setup and results. Finally, Section 6 discusses the conclusion and future work.

II. Literature Review

Text Classification (TC) is considered as a form of supervised learning task which involves assigning documents with predefined category labels depending on the suggested likelihood by a set of labeled training documents. Before the ML approach emerged in TC, knowledge engineering was the most common way in which the expertise of those working in the field was utilized. A set of rules was manually created into document classification within predefined categories [6]

explained that ML recruitment in TC presented several advantages as it lower cost and time in terms of the expert workforce only without any effect on precision. TC problems often present as a group of D documents and a group of S predefined categories with the significant purpose of assigning each (d_i, c_j) pair with a Boolean value ($d_i = \in D$ and $c_j = \in S$). An indeed given (d_i, c_j) value represents the decision of allocating document d_i to class c_j; however, a falsely assigned (d_i, c_j) value represents otherwise. Formally speaking, the primary task is the approximation of the obscure objective function $f: D \times S \rightarrow \{True, False\}$ [4], which clarifies an actual method of classifying the documents through the classification function $f': D \times S \rightarrow \{True, False\}$ in a manner there will be a minimal number of decisions off and f' that do not correspond.

Several learning algorithms have been developed and applied to TC. Among them includes k-Nearest Neighbor (KNN), Decision Tree (DT), Neural Networks, and Naive Bayesian (NB) [7-10].

The author in [11] compared these learning techniques and concluded that they all perform equally when there are more than 300 documents in each category. However, DT, KNN, and LLSF performed better than neural networks and Naïve Bayes Classification (NB) when there are fewer than ten positive training documents in each category.

Although several supervised ML approaches have been applied to TC, the task still demands the extra effort to predefine the categories and assign category labels to the training set documents. In large and dynamic text databases, this can be a complicated task. According to [6], inter-indexer inconsistency is another phenomenon that makes TC complicated. Accordingly, two experts may disagree on the category to designate a document. For example, a story about Bill Clinton and Monika Lewinsky could be classified under politics, gossip, both types, or neither category based on the subjective decision of human indexer [6]. The first synthesis that motivates us to discuss this is as follows:

- A large number of classification text.
- A large feature space.

TC's applications include document organization, hierarchical web page categorization, and text filtering. Document organization indicates the task of structuring documents into folders (maybe flat or hierarchical). For example, the incoming adverts to an editorial office may be grouped into categorized like Cars, Real Estate, Computers, etc., before publication. Conversely, text filtering indicates to the classification of a dynamic group of documents to relevant and irrelevant groups. This is illustrated in a news system in which articles in a newspaper are filtered from a news agency [6]. For instance, the delivery of news unrelated to sports in a sports newspaper will be blocked. Likewise, incoming messages may be classified as Arabic or not Arabic by a document filter in a bid to block Arabic message delivery [12]. The hierarchical classification process is one of the most flexible processes in the web browsing process due to the ease of navigating the hierarchical form of the categories and directing the search to a specific type instead of putting a general query on a search engine for general purposes. The hierarchical classification of documents generally requires the subdivision of the classification problem into smaller classification tasks. Some of the previous studies that addressed hierarchical document classification are [13-16].

TC using ML techniques entails a preprocessing of the texts, which require the transformation of the document into suitable forms for applying the learning algorithms. [17] introduced the commonly used vector space model called document representation. This model represents each document as a vector with each separate dimension corresponding to the word distinct occurring for all the words in the document. Text classifiers are applied for crime detection and weather prediction. In addition, they are also used to detect and track Arabic over documents. Table I showed the Arabic classification process is both detailed and benchmarked with previous works.

TABLE I. THE PROCESS OF ARABIC CLASSIFICATION

Training Phase	**Testing Phase**
Language preprocessing	***Language preprocessing***
Stemming and stop word removal, Tokenization, Normalization.	Stemming and stop word removal, Tokenization, Normalization.
Removal	***Removal***
Collection of Word	Collection of Word
Representation	***Representation***
TFIDF	TFIDF
Classifier	Classifier
Naïve Bayesian	Naïve Bayesian
Evaluation Accuracy	Evaluation Accuracy

III. BACKGROUND OF STUDY

A. Document Preprocessing

As indicated earlier, the scheme of document grouping in which intra-group similarities are high and low [18]. The essential process is preprocessing due to its significant role in improving and developing these schemes for any classifiers. The text analyst should take into account the relationship between preprocessing and similar measures on Document Classification. Performing preprocessing of documents is an essential process for classifying documents that are implemented by applying ML techniques.

Reference [18] pointed out that the significance of such stage in Document Classification is attributed to the presence of large numbers of unnecessary words present in the documents, many of which negatively influence classification rather than help in that. Using documents as a whole with unnecessary words is complicated. There are many of these words. The researchers themselves presented some non-essential utterances that may be found in the documents. The terms are classified into conjunctions, other grammatically based categories, particles, and that are typically employed without providing any assistance to the researcher in the classification of the document. In addition to some of the words presented by researchers like these words in the English language "ate, eaten, eat" and "الاكل , يؤكل , أكلت" in Arabic language, it is possible to decrease the number of the unique document words. Hence, we conclude that documents free of

unnecessary words are applied to them with appropriate prior treatment, and this would improve and increase the performance of the Document Classification approach.

After applying this step, the documents must be converted into a form appropriate for the representation process. Thus, the learning algorithms application is conducted. Then work will be done to remove unnecessary words like special markers and punctuation marks. To perform this process, many commonly used tasks, namely normalization, tokenization, stop-word removal, and mainly stemming, needed to be done. These tasks will be illustrated below, according to the review of previous studies.

B. Classification Algorithm

In the supervised algorithms, it is assumed that the categorical structure of a given database is already known. The supervised algorithms require a set of labeled documents to map documents into the predefined labels. As mentioned previously, it is challenging to determine the category and correct label of the training sets, particularly in large databases. Hence, this section will focus on the commonest supervised algorithm, NB.

NB is one of the common ML techniques. It depends on the Bayes' theorem, which claims to have strong (naive) and cumulative independence assumptions. A thorough description of the fundamental probability model theorem serves as the independent feature for the NB. A proper NB classifier could simply presume that there is no relationship between the existence or absence of a given class feature with that of any other feature. Also, and considers a simple probabilistic-based classifier. This assumption is expressed as follows:

$$\mathrm{P}(C_i|d) = \frac{P(C_i)P(d|C_i)}{P(d)} \quad (1)$$

Where $P(\mathrm{C_i}\,|d)$ it indicates a previous possibility of category C_i in the presence of another instanced, $P(\mathrm{C_i})$ represents the possibility of category C_i which might be calculated through:

$$P(C_i) = \frac{N_i}{N} \quad (2)$$

where N_i = the number of documents belonging to the category C_i, and N represents categories number, $P(d|\mathrm{C_i})$ represent the possibility of d document belongs to the given category C_i, as well as P(d) is the possibility of instance d. clarifies the complete NB pseudocode.

- BEGIN: about all the available values.
- Follow the rules for every individual value as:
  - Calculate and count the values of the classes appearing.
  - Obtain the class, which is frequently occurring.
  - Make the rule, which connects this particular class with instance values.
  - Find out the rate at which the error occurred for the rule.
  - Choose the rules with the minor error rate END.

IV. Experiment Process

A. DataSet

The dataset includes a set specifically designed to assess the extraction of the Arabic text for Arabic classifiers created as part of TREC 2001. The group has 383,872 Arabic documents, mostly newswire dispatches issued by Agence France Press (AFP) between 1994 and 2000 [19]. Ground truth and standard TREC queries have been created for such collection: 25 queries were considered part of TREC 2001 (Technology, 2001). The collection of queries has matching relevance judgments produced utilizing the pooling technique. Based on that, part of TREC 2001 is defined for classifiers as in Table II, which include four classes along with a group of documents (a gross of 800 documents).

B. Normalization

The steps of normalization ensure a specific characters' order that allows multiple variants. The reason behind the importance of data normalization for Arabic experiments is attributed to the fact that different encoding guidelines might either be used or not used at all by newspaper article and sometimes occurs in the same language. The following steps pinpoint the normalization of corpus and quires employed by [20]:

- Punctuation Removal.
- The removal of diacritics. Some entries consisted of weak vowels. Such elimination enabled text to become compatible.
- The removal special characters and numbers.
- Substituting آ, أ , and إ with ا.
- Substituting the end of ى with ي.
- Substituting the end of ة with ه.

C. The Removal of Stop-Words

Terms that usually frequently appeared in every document are so-called stop-words. These terms give no hint of their core document contents. Stop-words are determined so that a stop-words list could be established [21] . For that, an important thing in the Arabic system to omit stopping words from documents during preprocessing. Consequently, it is omitted from the group of indexed terms. Since there is no unified stop-words list of Arabic systems that can be used in classification systems. In this study, Khoja stop-words list tested with light10 stemmer [22].

TABLE II. Summary Description of Arabic Data Set

Categories	# of document
1	200
2	200
3	200
4	200

D. Tokenization

Converting a word into a distinctive word in text processing is a highly significant step. The tokenization process bears the responsibility of splitting the text into tokens, defining boundaries, words, numbers, and abbreviations. Arabic text tokenization is an important initial step as part of pre-processing phase. To define the complete word, in such a paper, the word was considered bound by white space marks to tokenize the Arabic text. Most importantly, the stemming process is regarded as the follow-up step after tokenization and the removal of stop-words.

E. Stemming

The term stemming indicates an approach of conflation that seeks to locate a common stem for a group of words in a text [23, 24]. For the conflation process, we used light10 stemmer by following the same process used by Larkey's [25]:

- They are deleting "و" ("and") if the rest of the word has greater than or equal to three letters. Regardless of the importance of deleting"و," it considers also problematic due to the fact that many popular Arabic words begin with this letter. Therefore, the length standard is more stringent here than the specific definite articles.
- Omitting every definite article, as this omission leaves word length greater than or equal to two letters.
- Dwelling into the list of suffixes once in the order (right to left) as shown in Table III, omitting every one of them that were at the end of the word if these omitting leaves word length greater than or equal to two letters.
- Table III shows the eliminated strings. Both conjunctions and definite articles are considered 'prefixes'. Light10 stemmer does not omit any character that can be considered an Arabic prefix.

TABLE III. PREFIXES AND SUFFIXES THAT ARE ELIMINATED THROUGHOUT LIGHT 10

Prefixes	Suffixes
ال، وال، بال، كال، فال، لل، و	ها، ان، ات، ون، ين، يه، ية، ه، ة، ي

F. Estimation

To complete estimation, Term Frequency (TF) x Inverse Document Frequency (IDF)) (TF x IDF) weighting was used for the weighting calculated as.

$$w_i = tf_i . log\left(\frac{N}{df}\right) \quad (3)$$

In classification problems, the estimation measures are generally determined from a matrix by employing a group of incorrectly and correctly categorized for each class (called the confusion matrix). Table IV revealed the confusion matrix for a binary categorization problem with classes that are merely positive and negative.

Here, FP, FN, TP, and TN are described thus:

- False Positives (FP): The negative examples are wrongly projected as positive.
- False Negatives (FN): positive instances which are wrongly predicted as negative.
- True Positives (TP): The positive examples that are correctly projected as positive.
- True Negatives (TN): The negative instances which are correctly predicted as negative.

The accuracy rate (ACC) is the commonly used estimation measure on the ground that estimates the classifier efficiency depends on its proportion of correct projections. The ACC of a classifier is calculated as follows:

$$ACC = ((TP + TN)/(TP + TN + FP + FN)) * 100 \quad (4)$$

TABLE IV. CONFUSION MATRIX

Projected Class		
True Class	***Positive***	***Negative***
Positive	TP	FN
Negative	FP	TN

V. EXPERIMENTAL SETUP AND RESULTS

As soon as the documents of the text are processed, they go through the classification tasks that occur by both stemming and converting them to an appropriate format. The collection of the document is classified twice for evaluation objectives. The first categorization is without stemmer, in which document collection occurs before stemming and applies to documents, whereas the second categorization is referred to as light stemmer that is used for the document collection. In respect of the available classifier approach, the overall dataset is classified for cross-validation. In conclusion, the cross-validations are classified as well as those in [26] for estimation. The remaining folds are employed for testing objectives, whereas K-fold cross-validation, K - 5 folds are used for validation and training.

Table V reveals the findings of the classifier accuracy by employing stemmer and without stemmer as preprocessing. It reveals that the accuracy resulted from the light stemmer was better than the classifier without stemmer detection, which the NB with light stemmer got 35.0745 higher than the NB 33.831% without stemmer. After contrasting them, the stemmer got better accuracy than the classifier. The last evaluation by employing the number of features revealed that without stemmer worse in decreasing the number of features, whereas stemmer was better, as shown in Table V.

TABLE V. EFFECT OF THE LIGHT STEMMER AND WITHOUT STEMMER USING NB AS A CLASSIFIER

Stemmer	Without Stemmer	Light Stemmer
Accuracy	33.830846	35.074627
#features	91756	46167

VI. Conclusion and Future Work

The experiments revealed that applied stemmer as preprocessing on our data set considers significant. Therefore, it has a substantial impact on the NB classifier on side accuracy and features number. Without stemmer, it does not significantly impact our dataset, as illustrated in Fig. 1 and 2. Major weakness concerning the without stemmer in preprocessing pertains to the dimensionality of terms requiring the second contribution to fill the huge term called feature selection. The critical concepts to be taken from the web to improve Arabic preprocessing are indicated. Therefore, the real datasets are employed in the experiments of this study. The primary finding regarding stemmer is essential for categorization. As stated earlier, it is better than with stemmer. In general, the finding is inconclusive due to the effect of classification or the chosen features revealed.

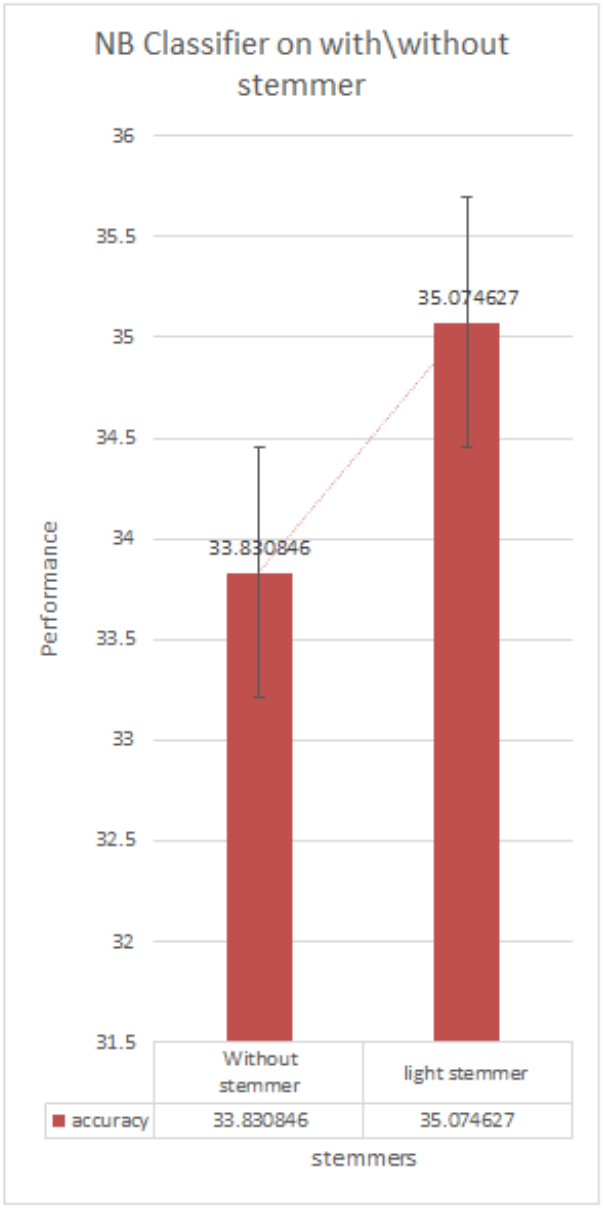

Fig. 1. Performance of Arabic Classifier using with/without Stemmer.

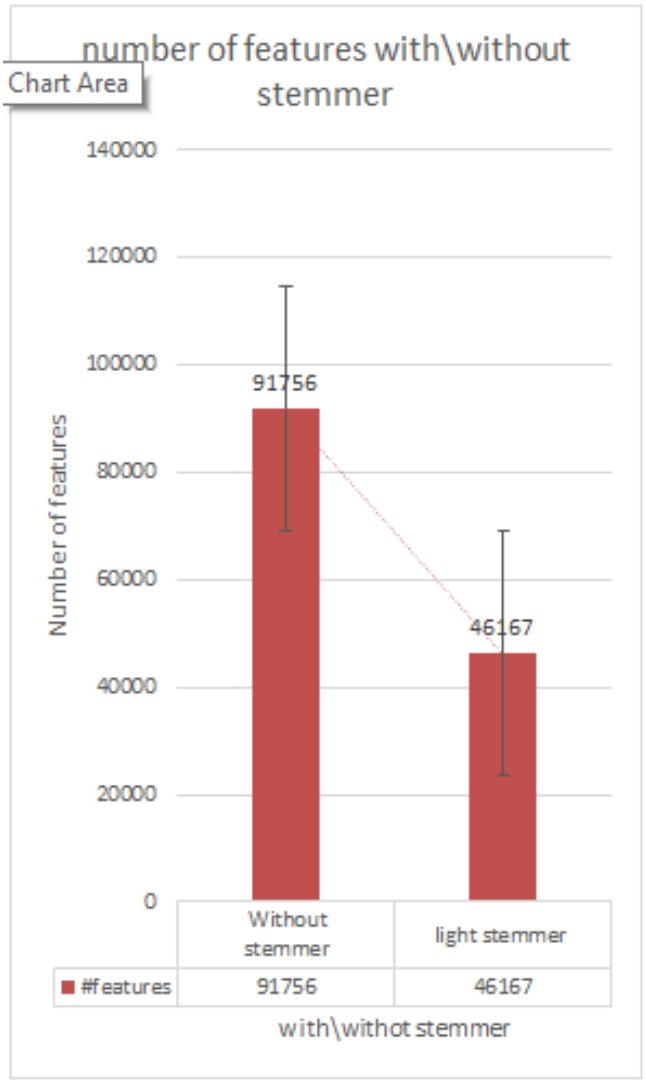

Fig. 2. Effect of Number of Features using with/without Stemmer.

Nevertheless, the central related weakness without stemmer in preprocessing is the dimensionality of terms revealed, which requires future work to bridge the gap concerning huge terms number called feature selection. This study answers the question regarding the effects of classifiers algorithms on Arabic classifiers with/without employing stemming of words and the purpose of investigating the performance of the used NB as Arabic classifiers on the classification performance with/without the employment stemming of terms.

A lot of open questions of the study are unanswered. Content classification considers a significant research area that provides various directions for additional studies. As a result, this section presents some suggestions for future research. Future studies conducted in this field are recommended to read this paper to try different classifiers and contrast their performance by employing various stemmer's algorithms. Also, using feature selection strategies for reducing the dimensionality of terms in the datasets as well as choosing the best features depending on their relevancy and significance to the subject class.

ACKNOWLEDGMENT

The researchers wish to thank (LDC) the Linguistic Data Consortium for their supplement of LDC2001T55 Arabic Newswire Part 1 at no cost, as one of the students of the Fall 2012 LDC Data Scholarship program.

REFERENCES

[1] E. G. Dada, J. S. Bassi, H. Chiroma, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," Heliyon, vol. 5, no. 6, p. e01802, 2019.

[2] G. Jain, M. Sharma, and B. Agarwal, "Spam detection on social media using semantic convolutional neural network," International Journal of Knowledge Discovery in Bioinformatics (IJKDB), vol. 8, no. 1, pp. 12-26, 2018.

[3] H. Gupta, M. S. Jamal, S. Madisetty, and M. S. Desarkar, "A framework for real-time spam detection in Twitter," in 2018 10th International Conference on Communication Systems & Networks (COMSNETS), 2018: IEEE, pp. 380-383.

[4] M. Revanasiddappa and B. Harish, "A new feature selection method based on intuitionistic fuzzy entropy to categorize text documents," IJIMAI, vol. 5, no. 3, pp. 106-117, 2018.

[5] J. A. Zdziarski, Ending spam: Bayesian content filtering and the art of statistical language classification. No starch press, 2005.

[6] F. Sebastiani, "Machine learning in automated text categorization," ACM computing surveys (CSUR), vol. 34, no. 1, pp. 1-47, 2002.

[7] T. Joachims, "Making large-scale SVM learning practical," Technical Report, 1998.

[8] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," Stanford InfoLab, 1997.

[9] B. Masand, G. Linoff, and D. Waltz, "Classifying news stories using memory based reasoning," in Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, 1992, pp. 59-65.

[10] Y. Yang, "An evaluation of statistical approaches to text categorization," Information retrieval, vol. 1, no. 1-2, pp. 69-90, 1999.

[11] Y. Yang and X. Liu, "A re-examination of text categorization methods," in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pp. 42-49.

[12] L. Özgür, "Adaptive Anti-Spam Filtering Based on Turkish Morphological Analysis, Artificial Neural Networks and Bayes Filtering," Bogazici University. Institute for Graduate Studies in Science and Engineering, 2003.

[13] M. Sahami, "Using machine learning to improve information access," Stanford University, Department of Computer Science, 1998.

[14] S. Dumais and H. Chen, "Hierarchical classification of web content," in Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000, pp. 256-263.

[15] M. Wedyan, B. Alhadidi, and A. Alrabea, "The effect of using a thesaurus in Arabic information retrieval system," Int. J. Comput. Sci, vol. 9, pp. 431-435, 2012.

[16] G. Kanaan and M. Wedyan, "Constructing an automatic thesaurus to enhance Arabic information retrieval system," in The 2nd Jordanian International Conference on Computer Science and Engineering, JICCSE, 2006, pp. 89-97.

[17] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," Communications of the ACM, vol. 18, no. 11, pp. 613-620, 1975.

[18] W. B. Croft, D. Metzler, and T. Strohman, Search engines: Information retrieval in practice. Addison-Wesley Reading, 2010.

[19] A. Cole, D. Graff, and K. Walker, "Arabic Newswire Part 1 Corpus (1-58563-190-6)," Linguistic Data Consortium (LDC), 2001.

[20] J. Atwan, M. Mohd, H. Rashaideh, and G. Kanaan, "Semantically enhanced pseudo relevance feedback for arabic information retrieval," Journal of Information Science, vol. 42, no. 2, pp. 246-260, 2016.

[21] B. Alhadidi and M. Alwedyan, "Hybrid Stop-Word Removal Technique for Arabic Language," Egyptian Computer Science Journal, vol. 30, no. 1, pp. 35-38, 2008.

[22] J. Atwan, M. Mohd, and G. Kanaan, "Enhanced arabic information retrieval: Light stemming and stop words," in International Multi-Conference on Artificial Intelligence Technology, 2013: Springer, pp. 219-228.

[23] J. Atwan and M. Mohd, "Arabic Query Expansion: A Review," Asian Journal of Information Technology, vol. 16, no. 10, pp. 754-770, 2017.

[24] J. Atwan, M. Wedyan, and H. Al-Zoubi, "Arabic text light stemmer," Int. J. Comput. Acad. Res, vol. 8, no. 2, pp. 17-23, 2019.

[25] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Light stemming for Arabic information retrieval," in Arabic computational morphology: Springer, 2007, pp. 221-243.

[26] J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning (no. 10). Springer series in statistics New York, 2001.