EasyChair Preprint

№ 80

An Enhanced LSI based Search Engine for Arabic Medical Documents

Fawaz Al-Anzi and Dia Abuzeina

April 21, 2018

An Enhanced LSI based Search Engine for Arabic Medical Documents

Fawaz S. Al-Anzi
Department of Computer Engineering
Kuwait University
Kuwait City, Kuwait
fawaz.alanzi@ku.edu.kw

Dia AbuZeina
Department of Computer Engineering
Kuwait University
Kuwait City, Kuwait
abuzeina@ku.edu.kw

***Abstract*—vector space model (VSM) is widely used for representing text documents in data mining and information retrieval (IR) systems. However, this technique poses some challenges such as high dimensional space and semantic loss representation. Therefore, latent semantic indexing (LSI) proposed to reduce the feature dimensions and to generate semantic rich features that represent conceptual term-document associations. In particular, LSI successfully implemented in search engines and text classification tasks. In this paper, we propose a novel approach to enhance the standard LSI method based on cosine measures instead of words occurrences to form LSI term-by-document matrix. We empirically evaluated the performance using an Arabic medical data collection that contains 800 documents with 47,222 unique words. A testing set contains five medical keywords used to evaluate the quality of the top-20 retrieved documents using different singular values (i.e. different number of dimensions). The results shows that the performance of the proposed method outperforms the standard LSI.**

Keywords—Arabic Text, Latent Semantic Indexing, Search Engine, Dimensionality Reduction.

I. INTRODUCTION

Due to the explosive growth of online information, search engines have a prominent role in information retrieval (IR) and web mining applications. The web is the largest repository of public data that undoubtedly requires efficient algorithms for retrieving and filtering out the textual information as well as other objects type. Hence, search engines are becoming more intelligent in filtering desired content. In general, textual data is represented using vector space model (VSM) where each document is represented using a vector of components (i.e. attributes), many of which could be zero. However, SVM poses some challenges such as huge feature vectors and semantic loss representation. Therefore, Latent semantic indexing (LSI) technique proposed to alleviate such challenges and hopefully to enhance the performance. LSI aims at transforming the original textual vectors into conceptual vectors that characterized by two distinguished properties; reduced dimensions and semantic rich features. The intrinsic nature that determines the quality of LSI is the semantic property that inforce returning semantically close documents without the constraint to have the exact searching keywords.

LSI based on a theorem from linear algebra that called singular valued decomposition (SVD). The SVD can transform the textual data represented as a large term-by-document matrix into a smaller semantic space represented as three matrices where the product of the generated matrices equal the original term-by-document matrix. Hence, the first step of LSI is to decompose (i.e factorize) the term-by-document (A) matrix as follows: $A=USV^T$ where U is a matrix that provides the weights of terms, S provides the eigenvalues (also called singular values) for each principal component direction, and V^T is a matrix that provides the weights of documents. V^T is the matrix that contain the documents feature vectors that generally used in IR and text mining applications. Fig. 1 shows the decomposition process that truncates a term-by-document matrix (A) into three matrices.

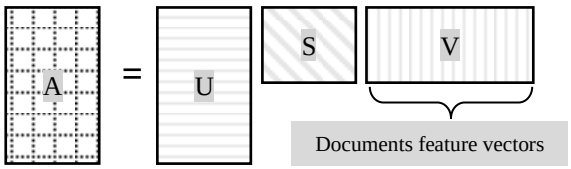

Fig. 1. SVD Decomposition Process

Therefore, the standard implementation of LSI starts with a term-by-document matrix to generate the required feature vectors that used for classification. However, term-by-document matrix generally formed using different values such as (Boolean flags, counts, or weights) to keep track of the occurrences or non-occurrences of terms in documents. For classification, the generated LSI feature vectors are generally utilized using a similarity measure such as Euclidian, Mahalanobis, Manhattan , cosine similarity, etc. Cosine similarity measure is known to be a one of popular distance measures in the pattern recognition. In this paper, we propose using the cosine measure instead of the standard values. Hence, the proposed method forms the term-by-document matrix using the cosine measures before employing the SVD process.

In next section, we present the motivation followed by the literature review in section 3. In section 4, we present the proposed method followed by the experimental results in section 5. We conclude in section 6.

This work is supported by Kuwait Foundation of Advancement of Science (KFAS), Research Grant Number P11418EO01 and Kuwait University Research Administration Research Project Number EO06/12.

II. MOTIVATION

Text mining systems are intuitively in need for very efficient algorithms that intelligently understand the search engine's documents as well as the query's keywords. Moreover, the huge online data requires considering the semantic relationships between the words and the documents that is also called co-occurrences. Unlike trivial searching method that based on traditional text matching, LSI characterized by semantic rich that enables the system to return useful results without having exact match words between the document and the query's keywords. For example, if we search for the word "coffee", it is expected that the system to return many documents related to this word, however, it might return other related documents that have no "coffee" word, but semantically related to the word "coffee". That is, it is possible to obtain documents belong to topics such as stimulant effects, caffeine, etc. Fig.2 shows an example of an article that contains "coffee" word. The figure also shows that the document has other words such as "skin". Therefore, the searching process for the word "coffee" might return documents related to "nervous system" topics that, on the same time, has no "coffee" word. This is the power of the LSI method and one reason why it is so popular.

إدمان **القهوة (coffee)** هل هو أمر حقيقي؟
بينت عدة دراسات ان نظرية ادمان الجسم على **القهوة (coffee)** فيه جانب من الصحة، اعتمادا على معنى الادمان. فمادة الكفايين من المواد المنبهة والمحفزة **للجهاز العصبي (nervous system)** الرئيسي (الدماغ) وتناولها بشكل مستمر يسبب تعود الجسم عليها. لكن ليس لمادة الكافايين أثر خطير يهدد الصحة الجسدية او النفسية أو الاجتماعية او حتى المالية، مثلما يتسبب به التعود على تناول العقاقير او المخدرات. رغم ان كثرة شراء المشروبات الكافينية من المقاهي بشكل يومي يكلف مالا. وان كنت ممن تعودوا على تناول كوب او عدة اكواب من **القهوة (coffee)** يوميا فإن توقفك عن تناولها ليوم سيسبب ظهور عدة اعراض نتيجة انسحاب مادة الكافايين من الجسم مثل: الصداع، التوتر، العصبية والكآبة وتعكر المزاج وصعوبة التركيز. لكنها لا تعتبر أعراضا مؤلمة او تسبب سلوكيات ضارة تدفعك الى أذية النفس والاخرين او الهيجان او ارتكاب الجرائم. ونتيجة لكل ما ذكر، لا يعتبر كثير من الخبراء تعود الجسم على مادة الكافايين إدمانا من النوع الجدي.

Fig. 2. An Example of words co-occurance of "coffee" and "nervous system"

Enhancing searching process thorough overwhelming digital data requires endless research effort to adequately satisfy users' requests. In fact, text mining is a challenge task since documents usually have mixed contents that make it difficult to digitally understand the document's category. For illustration, Fig. 2 shows a document that has different words such as "headaches" <> "صداع", "addiction" <> "ادمان", "drugs" <> "مخدرات", "tension" <> "التوتر", "frenzy" <> "الهيجان", and "crimes" <> "الجرائم". Such words diversity make it vague for IR algorithms. By nature, medical documents require precise algorithms that adequately find the proper documents for the user.

The LSI is proven to be a valuable tool that reveals the semantical relations between the data objects (i.e the words, in this research). Based on detected underlying semantic distinctions, LSI will be able to bring out the relevant documents that do not contain the searching keyword at all. Fig. 3 shows a medical article that is related to "التنفس"<>"breathing". If a user search for the "الاوكسجين" <> "oxygen" word, then the semantic loss methods (i.e plain keyword search) will fail since there is no exact match between the searching word and the document's words. However, LSI does support the semantic search and this is what the users look for especially for medical documents.

الغُبار قد يضرّ المصابين بمشاكل في **التنفّس (breathing)**

جنيف – رويترز – قالت منظمة الصحة العالمية ان الغبار الناجم عن ثورة بركان ايسلندا قد تضر أيضا الاشخاص الذين يعانون مشاكل في التنفس (**breathing**)، لأن هذه الجزيئات عند استنشاقها يمكن ان تصل الى المناطق المحيطة من القصيبات التنفسية (**breathing**) والرئتين (**lungs**)، ويمكن أن تسبب مشاكل، خاصة للأشخاص الذين يعانون الربو أو مشاكل بالجهاز التنفسي (**breathing**).
وفي جانب متصل، أكدت الهيئة البريطانية للوقاية الصحية أن الرماد البركاني لا يشكّل خطورة كبيرة على الصحة العامة، ومن غير المرجّح أن يسبب ضررا كبيرا، حيث يتطلب الامر التعرض بشكل كبير جدا للغبار المنخفض السَّمية حتى يكون هناك تأثير في الناس.
وقال كين دونالدسون أستاذ علم السموم التنفسية (**breathing**) في جامعة أدنبره لـــ «رويترز»: «هناك تأثير ضعيف بشكل كبير في الغلاف الجوي، حيث يتشتت بفعل الرياح، ما يعني أن الكمية التي تصل إلى الأرض صغيرة للغاية». واتفق دونالدسون على ان الناس المصابين بأمراض بالرئة (**lung**) بالفعل يجب ان يبقوا في اماكن مغلقة إذا كان هناك تغيير ملموس فى مستويات الجسيمات.

Fig. 3. An Example of breathing medical document

III. LITERATURE REVIEW

In the literature, there are many studies discuss LSI technique. In particular, LSI used for text mining task such as text classification, text summarization, text clustering, search engines, etc. LSI initially presented by Deerwesterin [1] as a standard dimension reduction techniques in IR. Reference [2] presented an algorithm to enhance the results of search engines. The algorithm combines common phrase discovery and LSI techniques to separate search results into meaningful groups. Reference [3] presented a new implementation of the standard LSI. The new implementation aims at providing efficient, extensible, portable, and maintainable LSI. Reference [4] presented a theoretical model for understanding the performance of LSI in retrieval applications. Reference [5] presented LSI based method for fully automated cross-language document retrieval in which no query translation is required.

Reference [6] described a word clustering approach based on LSI. Reference [7] proposed a local LSI method called "Local Relevancy Weighted LSI" to improve text classification by performing a separate SVD on the transformed local region of each class. Reference [8] used LSI to automatically identify conceptual gene relationships from titles and abstracts in a database citation. Reference [9] proposed and empirically

tested the feasibility and utility of post-retrieval clustering of digital forensic text string search results – specifically by using Kohonen Self-Organizing Maps (SOM) as a self-organizing neural network approach. Reference [10] proposed a hybrid term frequency – inverse document frequency (TF-IDF) based algorithm and a clustering based algorithm for obtaining multi-post summaries of Twitter posts along with detailed analysis of Twitter post domain. Reference [11] used LSI for automatic software clustering. They used LSI as the basis to cluster software components, source code and its accompanying documentation. Reference [12] proposed two text summarization approaches: modified corpus-based approach (MCBA) and LSI-based approach.

LSI has been widely documented a retrieval method that employs SVD for semantic rich reduced feature vectors. Nevertheless, utilizing LSI and SVD requires understanding which values in the reduced dimensional space contain the words relationships (latent semantic) information. Hence, many studies in the literature discussed this important aspect. Reference [13] presented an empirical study of the required dimensionality for large-scale LSI applications. Reference [14] developed a model for understanding which values in the reduced dimensional space contain the term relationship (latent semantic) information.

Regarding cosine similarity, it is well-known similarity measure that has widely mentioned in the literature. Reference [15] indicated that cosine similarity dominants similarity measures in IR and text classification. This measure based on the the cosine of the angle between two vectors. Reference [16] demonstrated that the similarity between two documents can be measured using the cosine of the angle between the two document feature vectors represented using VSM. Theodoridis and Koutroumbas in [17] defines cosine similarity measure as: $Scosine(x,y) = x^T y/\|x\|\|y\|$ where $\|x\|$ and $\|y\|$ are the lengths of the vectors x and y, respectively. Reference [18] presented that cosine similarity is a robust metric for scoring the similarity between two strings. Reference [19] demonstrated that cosine similarity is used to find vectors neighbourhood. Reference [20] demonstrated that cosine similarity is easy to interpret and simple to compute for sparse vectors, they indicated that it is widely used in text mining and IR. Reference [21] used cosine similarity measure for Arabic language text summarization. Reference [22] used cosine measure in the language identification problem.

IV. The Proposed Method

This work include two parts. In the first part, we evaluate the search engine performance using the standard LSI technique. In the second part, we propose an enhanced method for the standard LSI. Initially, the preprocessing step performed by declaring the stop words and ignore characters list. In addition, all small words that are less than three characters length are discarded. A normalization process also performed to change some Arabic characters such as ($^{i}\rightarrow^{ا}$). As shown in Fig. 4, the term-by-document (A) matrix created using the unique words in the used corpus. The term-by-document (A) matrix weighted using TF-IDF. For comparison purpose, the standard LSI used by decomposing A into three matrices (U: Term by dimension, S: Singular values, and VT: Document by dimension). The diagonal of matrix S contains the singular values that one can choose as the desired reduced dimensions. In general, not all singular values are taken; instead, only the most important values are considered starting from the first singular values up to the desired value (k).

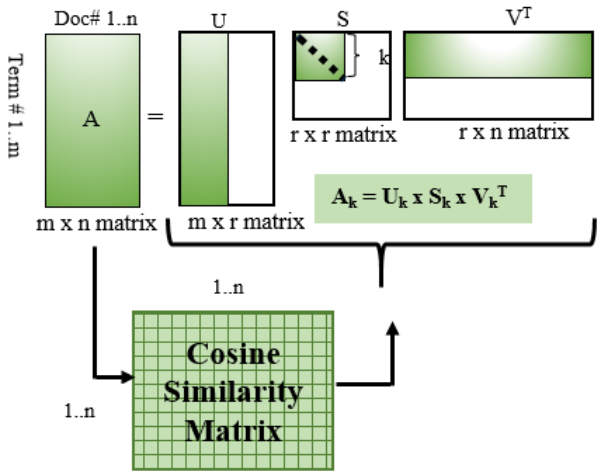

Fig. 4. Truncation of SVD for LSI and forming cosine similarities matrix

The proposed method has an extension of the standard LSI by creating a new matrix called cosine similarity matrix. This new matrix has the cosine similarities between all documents in the corpus instead of the co-occurrences (i.e. instead of the frequency of a word in a document) that usually used when creating term-by-document matrices. Hence, the enhanced method summarized using four main steps as follows: creating the standard term-by-document matrix using words co-occurrences. The matrix weighted using TF-IDF. Using the standard term-by-document matrix, a new matrix called cosine similarity matrix created using the document feature vectors in the standard term-by-document. Finally, the cosine similarity matrix is decomposed suing SVD to generate the enhanced feature vectors that used in the search engine. Of course, Different singular values (k) has to be investigated to find the optimal performance.

Fig. 5 demonstrates an example how to create a cosine similarity matrix for three documents. The diagonal entries contain 1.0 as the cosine of is 1.0 zero (i.e. the document with itself). Hence, as our corpus contains 800 documents, the cosine similarity matrix of the used corpus is of size 800 × 800. Of course, the cosine similarity matrix is symmetric matrix.

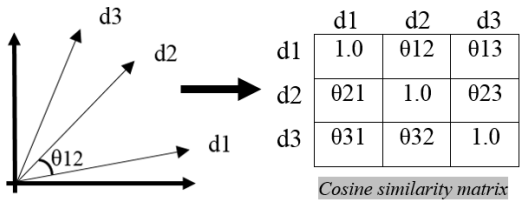

Fig. 5. A matrix of cosine similarities for three vectors

In both cases, the standard LSI or the proposed method, the query's keywords have to be mapped to the LSI space. For the standard LSI, the query's feature vectors have to be transformed into the new reduced space that is called "folding-

in". This is done using the following formula: $V'^{T}=AUS^{-1}$. Hence, V'^{T} contains the reduced query's feature vectors that are used along with V^{T} in the classification process. For the proposed method, the query's feature vectors has two transformation steps. The first is regarding the cosine measures against all feature vectors of the training data before using the folding-in technique as a second step (i.e. like the standard LSI but for the cosine measure instead of the words co-occurrences). Fig. 6 shows how to generate the query's vector in terms of cosine similarity. Hence, the cosine similarity matrices of the training and the testing set are generated to be used with SVD.

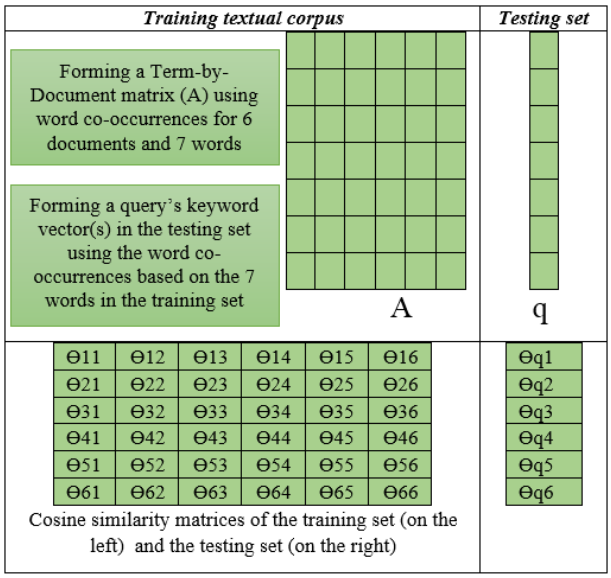

Fig. 6. Forming the cosine similiryt matrices of the training and testing sets

V. The Experimental Results

The proposed method evaluated using an Arabic textual corpus contains 800 documents, 353,888 words, and 47,222 unique words. The data collection is regarding medical stories obtained from Alqabas [23] Kuwaiti newspaper. A testing set contains five medical keywords used as queries for the developed search engine. Hence, the testing set arbitrary contains {"الزهايمر" <> "Alzheimer", "فيروس" <> "virus", "الاوكسجين" <> "oxygen", "القهوة" <> "coffee", "اشعة" <> "rays"}. However, a query could have more than one word .Table I shows more information regarding the testing set and its appearance in the training corpus. Table I shows that the word "القهوة" <> "coffee" has appeared 143 times in 36 different documents.

TABLE I. Testing set information

Query word	Total appearance	Total documents
"الزهايمر" <> "**Alzheimer**"	32	14
"فيروس" <> "**virus**"	204	66
"الاوكسجين" <> "**oxygen**"	55	36
"القهوة" <> "**coffee**"	143	36
"اشعة" <> "**rays**"	324	103

Since the number of singular values is important in LSI applications. We consider a wide range of singular values to measure the performance for both cases (i.e. standard LSI and the proposed method). Hence, the search engine was evaluated using different number of feature vectors dimensions. That is, a series of experiments performed using the following k :{ k=10, 20, 30, 40, 50, 60, 70, 80, 90, 100,150,200, 250,300, 350,400, 500}. At each singular value, we analyzed the top-20 retrieved documents to investigate the query's keyword occurrences.

Table II shows the performance of the "الزهايمر" <> "Alzheimer" word. In the table, the first row indicates the medical keyword we searched for. The first column indicates the singular values k that starts at 10 and ends at 500. At k=10, the word "الزهايمر" <> "Alzheimer" found at the first document zero times while it was found in the top-20 retrieved document three times. On the other hand, it was found one time in the first document and 4 times in the top-20 of the proposed method. The results show that the retrieved document using the proposed method are of high quality compared to the standard LSI even with lower dimensions. For illustration, at k=80, the standard LSI retrieve a document that contains 1 occurrence of the searching word with 11 occurrences in the top-20 documents, while the proposed method return a document that contains 6 occurrences with 20 occurrences in the top-20 documents. Table II also shows that the maximum occurrences of the word "الزهايمر" <> "Alzheimer" is 27 times using standard LSI, however, it scored 29 occurrences using the proposed method. The results presented in Table II does not require the exact match cases as we considered the word "الزهايمر" <> "Alzheimer" is same as "زهايمر" and "بالزهايمر ", etc. Hence, different word's form are counted.

TABLE II. Searching Results for Different Dimensions of "Alzheimer"

"الزهايمر" <> "**Alzheimer**"				
	The Standard LSI		**The Proposed Method**	
k	***First doc.***	***Top-20 doc.***	***First doc.***	***Top-20 doc.***
10	0	3	1	4
20	0	6	6	13
30	0	7	6	13
40	0	10	6	18
50	1	3	6	18
60	1	7	6	18
70	1	11	6	18
80	1	11	6	20
90	1	7	6	16
100	1	11	6	16
150	1	19	6	21
200	1	21	6	23
250	1	18	6	23
300	1	18	6	25
350	3	24	6	25
400	3	**27** (max)	6	26
500	3	25	6	**29** (max)

Table III shows the performance of the "فيروس" <> "virus" word. Using k=40, the proposed method returns 114 occurrences of the searching word while it return only 106 occurrences at k=150. Hence, with lower dimensions, the proposed method demonstrates better results. The proposed method also gives better results for the first retrieved document

as it has 19 occurrences of the searching word, while it has nothing related to the searching word in the standard LSI.

TABLE III. SEARCHING RESULTS FOR DIFFERENT DIMENSIONS OF "VIRUS"

"فيروس" <> "virus"				
	The Standard LSI		**The Proposed Method**	
k	***First doc.***	***Top-20 doc.***	***First doc.***	***Top-20 doc.***
10	0	36	19	74
20	14	77	19	81
30	14	57	19	99
40	19	68	15	**114** (max)
50	19	74	19	92
60	19	79	15	87
70	19	82	19	91
80	19	94	19	94
90	19	91	19	96
100	19	78	19	96
150	19	**106** (max)	19	96
200	19	99	15	95
250	19	91	15	91
300	19	91	15	100
350	19	87	15	101
400	19	89	15	92
500	15	94	15	83

Table IV shows the performance of the "الاوكسجين" <> "oxygen" word. This word did not appear in first retrieved document for both the standard LSI and the proposed method. However, for the top-20 list, the proposed method scored 29 occurrences of this word while it was just 17 occurrences using standard LSI.

TABLE IV. SEARCHING RESULTS FOR DIFFERENT DIMENSIONS OF "OXYGEN"

"الاوكسجين" <> "oxygen"				
	The Standard LSI		**The Proposed Method**	
k	***First doc.***	***Top-20 doc.***	***First doc.***	***Top-20 doc.***
10	0	2	0	7
20	0	7	0	8
30	2	10	0	11
40	2	11	0	10
50	2	5	6	11
60	2	6	6	13
70	2	10	6	13
80	2	10	2	13
90	2	8	2	13
100	2	9	2	13
150	2	14	2	**18**
200	2	8	2	14
250	2	9	2	17
300	2	16	2	16
350	2	16	2	**19**
400	2	**17** (max)	2	**19**
500	2	17	2	**26** (max)

Table V shows that the first document returns 60 occurrences (it is relatively long document) of the word "القهوة" <> "coffee", while it does not return the query word in the first document using the standard LSI. It is worthwhile observation that the standard LSI retrieved this long document at k=90 while it is retrieved at k=10 using the proposed method.

TABLE V. SEARCHING RESULTS FOR DIFFERENT DIMENSIONS OF "COFFEE"

"القهوة" <> "coffee"				
	The Standard LSI		**The Proposed Method**	
k	***First doc.***	***Top-20 doc.***	***First doc.***	***Top-20 doc.***
10	0	68	60	60
20	8	80	60	88
30	8	84	60	93
40	60	84	60	94
50	8	85	60	105
60	8	85	60	105
70	8	95	60	111
80	8	95	60	110
90	60	105	60	115
100	60	107	60	116
150	60	105	60	119
200	60	111	60	**121**
250	60	114	60	**122** (max)
300	60	118	60	**122** (max)
350	60	117	60	**122** (max)
400	60	116	60	**122** (max)
500	60	**120** (max)	60	119

Table VI shows that the first document returns three occurrences of the word "اشعة" <> "rays" using both methods. Table VI also shows that the performance started decreasing after k=200. Therefore, each LSI based application has a particular range of singular values (k) where it gives the optimal performance.

TABLE VI. SEARCHING RESULTS FOR DIFFERENT DIMENSIONS OF "RAYS"

"اشعة" <> "rays"				
	The Standard LSI		**The Proposed Method**	
k	***First doc.***	***Top-20 doc.***	***First doc.***	***Top-20 doc.***
10	3	64	3	40
20	3	39	4	108
30	2	34	13	104
40	3	56	4	104
50	10	101	5	105
60	8	104	8	**113**
70	8	93	8	112
80	8	93	8	112
90	8	100	8	112
100	8	100	8	**114**
150	8	**112** (max)	22	**116** (max)
200	8	108	22	111
250	22	110	22	108
300	10	105	22	106
350	10	104	22	102
400	10	105	22	108
500	2	100	22	97

In addition, we evaluated the performance by measuring the percentage of the returning matched words among all occurrences in the training set. For example, the word "اشعة" <> "rays" appears 324 times in the corpus. The standard LSI return this word 112 times in the top-20 list as indicated in table VI. However, the proposed method returns it 116 times. Hence, the percentage for the standard LSI is 112/324=0.346. For the proposed method, the percentage is 116/324=0.358. These percentages shown in table VII for all words of the testing set. The table also shows that the average of the percentages for the standard LSI is 0.571 and for the proposed method is 0.629.

This means that the proposed method outperform the standard LSI by 5.83% for the top-20 retrieved documents.

TABLE VII. THE PECENTAGE OF THE RETRIVED SERCHING WORDS

#	Word	The Standard LSI	The Proposed Method
1	"الزهايمر" <> "Alzheimer"	0.844	0.906
2	"فيروس" <> "virus"	0.520	0.559
3	"الاوكسجين" <> "oxygen"	0.309	0.473
4	"القهوة" <> "coffee"	0.839	0.853
5	"اشعة" <> "rays"	0.346	0.358
	Average	**0.571**	**0.629**

In fact, other evaluation methods are required since we only compare the match word while the semantic quality of the retrieved documents should also be evaluated. Fig. 7 shows the graphical representation of the performance differences between the standard LSI and the proposed method. The graph information is based on the percentages calculated in Table VII.

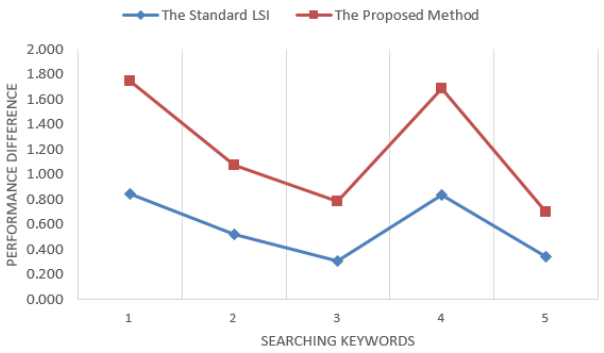

Fig. 7. The performance ehhancement using the prposed method

Finally, the proposed method is suitable for relatively small data collections. However, it might be not efficient for very large corpora that contains millions of documents. In fact, creating the cosine similarity matrix is $O(n^2)$ where n is the total number of documents in the corpus. Nevertheless, this method shows a possible enhancement especially when we look for precis results for highly mixed contents as medical documents.

VI. CONCLUSION

This paper discusses the LSI technique for search engines. We evaluated the performance of the standard LSI and an enhanced method that based on cosine similarity measure. The results shows that using the cosine similarities instead of word co-occurrences enhances the performance of search engines. The proposed method shows that the top-20 retrieved document are of more quality than the top-20 list retrieved using the standard LSI. As a future work, we propose to investigate the proposed method for larger data collections as well as investigating the time and space complexity of the proposed method. In addition, the evaluation should include the semantic quality instead only the plain matched words.

REFERENCES

[1] Deerwester, Scott, et al. "Indexing by latent semantic analysis." Journal of the American society for information science 41.6 (1990): 391.

[2] Osiński, Stanislaw, and Dawid Weiss. "A concept-driven algorithm for clustering search results." Intelligent Systems, IEEE 20.3 (2005): 48-54.

[3] Letsche, Todd A., and Michael W. Berry. "Large-scale information retrieval with latent semantic indexing." *Information sciences* 100.1 (1997): 105-137.

[4] Kontostathis, April, and William M. Pottenger. "A framework for understanding Latent Semantic Indexing (LSI) performance." Information Processing & Management 42.1 (2006): 56-73.

[5] Dumais, Susan T., et al. "Automatic cross-language retrieval using latent semantic indexing." *AAAI spring symposium on cross-language text and speech retrieval*. Vol. 15. 1997.

[6] Bellegarda, J. R., Butzberger, J. W., Chow, Y. L., Coccaro, N. B., & Naik, D. (1996, May). A novel word clustering algorithm based on latent semantic analysis. In Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on (Vol. 1, pp. 172-175). IEEE.

[7] Liu, T., Chen, Z., Zhang, B., Ma, W. Y., & Wu, G. (2004, November). Improving text classification using local latent semantic indexing. In Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on (pp. 162-169). IEEE.

[8] Homayouni, R., Heinrich, K., Wei, L., & Berry, M. W. (2005). Gene clustering by latent semantic indexing of MEDLINE abstracts. Bioinformatics, 21(1), 104-115.

[9] Beebe, Nicole Lang, and Jan Guynes Clark. "Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results." Digital investigation 4 (2007): 49-54.

[10] Inouye, David, and Jugal K. Kalita. "Comparing twitter summarization algorithms for multiple post summaries." Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on. IEEE, 2011.

[11] Maletic, Jonathan, and Naveen Valluri. "Automatic software clustering via latent semantic analysis." Automated Software Engineering, 1999. 14th IEEE International Conference on.. IEEE, 1999.

[12] Yeh, J. Y., Ke, H. R., Yang, W. P., & Meng, I. H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. Information processing & management, 41(1), 75-95.

[13] Bradford, Roger B. "An empirical study of required dimensionality for large-scale latent semantic indexing applications." Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008.

[14] Kontostathis, April. "Essential dimensions of latent semantic indexing (lsi)."System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on. IEEE, 2007.

[15] Elberrichi, Z., Rahmoun, A., & Bentaallah, M. A. (2008). Using WordNet for Text Categorization. Int. Arab J. Inf. Technol., 5(1), 16-24.

[16] Beil, Florian, Martin Ester, and Xiaowei Xu. "Frequent term-based text clustering." Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002.

[17] Theodoridis, S. and K. Koutroumbas (2008). Pattern Recognition, Fourth Edition, Academic Press.

[18] Tata, S., & Patel, J. M. (2007). Estimating the selectivity of tf-idf based cosine similarity predicates. ACM Sigmod Record, 36(2), 7-12.

[19] Rajan Chattamvelli, Data Mining Algorithms, Published by Alpha Science International Ltd., 2011

[20] Dhillon, I. S., & Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. Machine learning, 42(1-2), 143-175.

[21] Sobh, I., Darwish, N., & Fayek, M. (2006). A trainable Arabic Bayesian extractive generic text summarizer. In Proceedings of the Sixth Conference on Language Engineering ESLEC (pp. 49-154).

[22] Takçı, H., & Güngör, T. (2012). A high performance centroid-based classification approach for language identification. Pattern Recognition Letters,33(16), 2077-2084.

[23] Alqabas. (2016, October). Retrieved from http://www.alqabas.com.kw/Default.aspx