Reinforcing Arabic Language Text Clustering: Theory and Application

Fawaz S. Al-Anzi and Dia AbuZeina
Department of Computer Engineering, Kuwait University
fawaz.alanzi@ku.edu.kw; abuzeina@ku.edu.kw

Abstract:*This paper presents a novel approach for automatic Arabic text clustering. The proposed method combines two well-known information retrieval techniques that are latent semantic indexing (LSI) and cosine similarity measure. The standard LSI technique generates the textual feature vectors based on the words co-occurrences; however, the proposed method generates the feature vectors using the cosine measures between the documents. The goal is to obtain high quality textual clusters based on semantic rich features for the benefit of linguistic applications. The performance of the proposed method evaluated using an Arabic corpus that contains 1,000 documents belongs to 10 topics (100 documents for each topic).For clustering, we used expectation-maximization (EM) unsupervised clustering technique to cluster the corpus's documents for ten groups. The experimental results show that the proposed method outperforms the standard LSI method by about 15%.*

Keywords:*Arabic Text, Clustering,Latent Semantic Indexing,Latent Semantic Indexing, Expectation-Maximization.*

1. Introduction

With themassive amounts of textual information build up and the exponential growth of digital documents, there is a demand for better performing tools to aid the humans in handling and processing them. Text clustering is among the crucial linguistic techniques that has been widely investigated to enhance the information retrieval (IR) performance. Text clustering used in many linguistic applications such as search engines, topic detection, document organization, software clustering, gene clustering,text summarizations, etc.Reference [6]demonstrates that text clustering applied to several application domains such as organizing of results returned by search engines in response to a user's query, browsing large document collections, topic detection, and in generating a hierarchy of web documents.

In today's market, achieving user satisfaction within this astronomical growth of online data is becoming veryappealing to business investment. Search engines, e.g., Google and other high traffic query processing portals,are expected to meet and satisfy today's userdemands. As an example of such huge traffic, Table 1 shows the remarkable annual number of Google search queries [2].

Table 1. Annual number of Google search queries

Year	Annual Number of Google Search Queries	Average Search Queries Per Day
2014	2,095,100,000,000	5,740,000,000
2013	2,161,530,000,000	5,922,000,000
2012	1,873,910,000,000	5,134,000,000
2011	1,722,071,000,000	4,717,000,000
2010	1,324,670,000,000	3,627,000,000
2000	22,000,000,000	60,000,000
1998*	3,600,000	9,800

*Google's official first year of operation

Text clustering generally based on textual features that support semantic relationships between documents. Such semantic rich features can be generated using the latent semantic indexing (LSI) technique. LSI has another distinguished property that is a dimensionality reduction technique, which is very important when handling high dimensionality vectors as the case of vector space model (VSM) text representation technique. LSIis generally used with singular value decomposition (SVD) to generate the semantic feature vectors. Despite successful history of LSI technique, however, we found a potential performance enhancement through one more step of the standard LSI method. That is, the proposed step is to employ LSI one more time for the cosine similarity measures between all documents feature vectors (which sometimes called the weights of the documents). Clustering is then performed of the newly generated documents feature vectors. The clustering technique used in this paper is expectation-maximization (EM)that used to generate the category of each document.

In next section, we present theliterate review. In section 3, we present the theoretical background followed the proposed method in section 4. The resultsdiscussed in section 5 and the conclusion in section 6.

2. Literature Review

The literature has quite large studies that discuss text clustering problem. Since this study based on LSI, we highlight the relevant research related to this technique as follows:

- *Search Engines:*Reference [19] 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 [16] presented how LSI based text clustering can enhance the performance of search engines.
- *Software clustering:*Reference [18] used LSIfor automatic software clustering. They used LSI as the basis to cluster software components, source code and its accompanying documentation. Reference [15] introduced an LSI based semantic clusteringto group source artifacts that use similar vocabulary.
- *Semantic event detection:* Reference [28] presented an approach for sports video semantic event detection based on analysis and alignment of webcast text and broadcast video. They used probabilistic LSI and a conditional random field model (CRFM).
- *Generelationships detection:*Reference [13] used LSI to automatically identify conceptual gene relationships from titles and abstracts in a database citation.
- *Speech recognition:*Reference [5] used LSI to uncover the salient semantic relationships between words and documents in a given corpus. The relationships were extracted to be used with a statistical language model in a speech recognition engine. Reference [12] showed that using semantic information to generate mixture language models can outperform conventional single language model in large vocabulary speech recognition systems.
- *Text summarization:*Reference [26] proposed a multi-document summarization framework based on sentence-level semantic analysis and symmetric non-negative matrix factorization. Reference [29] proposed two text summarization approaches: modified corpus-based approach (MCBA) and LSI-based approach. Reference [21] demonstrated that text clustering can be used for finding the posts on a specific topic or to find various themes and sub-themes in microblog summarization.
- *Word clustering:* Reference [4] described a word clustering approach based on LSI.
- *Text classification:* Reference [17] proposed alocal 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 [2] employed LSI for Arabic text classification. Reference [30] used LSI and back-propagation neural network (BPNN) and modified back-propagation neural network (MBPNN) for text categorization.

3. Theoretical background

The proposed method utilize two techniques for text clustering that includes LSI and cosine similarity measure. The following are brief description of these two techniques:

- *Latent Semantic Indexing (LSI):* The first step of using LSI is forming a term-by-document matrix that is decomposed using SVD. Hence, SVD reduce the number of dimensions of a feature space without serious loss of information. SVD is based on a theorem from linear algebra which says that a rectangular m-by-n matrix A can be broken down into the product of three matrices - an orthogonal matrix U, a diagonal matrix S, and the transpose of an orthogonal matrix V. The theorem is usually presented as something like this:

$A_{mn} = U_{mr} S_{rr} V^T_{rn}$ *where r is the rank of A.*

Figure 1 demonstrates the reduced rank SVD of a matrix A. The black dashed bold line in the matrix S represents the values retained in computing rank k approximation. In general, not all singular values are considered. Instead, only the most important values starting from the first singular values up to the desired value. The reduced feature vectors of the matrix $V^T$are generally used for text clustering and text classification applications.

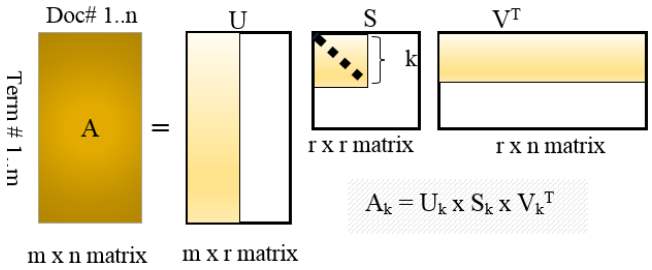


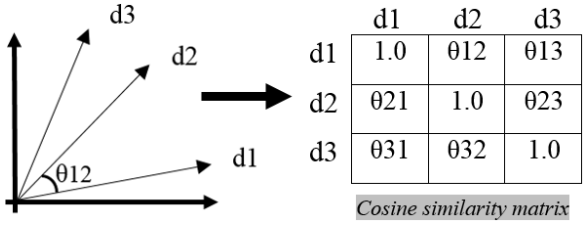
Figure 1.SVD representation of the document-term matrix

- *Cosine similarity measure:*Cosine similarity is a popular measure that has been extensively used in pattern recognition and text mining problems. Reference [9] indicated that cosine similarity dominant similarity measures in IR and text classification. This measure based on the the cosine of the angle between two vectors. Reference [3] demonstrated that the similarity between two documents can be measured using the cosine of the angle between the two vectors represented the documents in the VSM. Reference [25] defines cosine similarity measure as Scosine(x,y) = $\frac{x^T y}{\|x\|\|y\|}$where $\|x\|=\sqrt{\sum_{i=1}^{l} x_i^2}$ and $\|y\|=\sqrt{\sum_{i=1}^{l} y_i^2}$ are the lengths of the vectors x and y, respectively. Both $\boldsymbol{x}$ and $\boldsymbol{y}$ are $\boldsymbol{l}$-dimensional vectors. In related literature, we found

that the cosine similarity is a measure of similarity between two vectors, [10]. Reference [24] presented that cosine similarity is a robust metric for scoring the similarity between two strings. Reference [14] presented a simple textual example to indicate that cosine similarity is better than some other measure such as Euclidean distance. Reference [20] demonstrated that cosine similarity is used to find vectors neighbourhood. Reference [8] 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 [22] used cosine similarity measure for Arabic language text summarization. Reference [23] used cosine measure in the language identification problem.

4. The Proposed Method

The proposed approach includes employing LSI two times before clustering. The first time is the conventional LSI that is implemented to generate the features vectors using the frequency of the unique words in the corpus. Once obtaining the document features using the standard LSI, we create a new matrix that contains the cosine similarities between all documents. To clarify this method, suppose that we have three vectors and we need to create a matrix that contains the cosine similarities between these three vectors. We simply create a square matrix of 3x3 as shown in Figure 2. This figure shows three vectors that, supposedly, represent three document (d1, d2, and d3) and the cosine between all angles. The diagonal entries contain 1 as the cosine of 0 is 1. For example, the angle between d1 and d1 is 0, and the cosine of 0 is 1. The cosine similarity matrix is then decomposed using SVD. The idea is that LSI will generate features that are semantically related better than the standard LSI method.



	d1	d2	d3
d1	1.0	θ12	θ13
d2	θ21	1.0	θ23
d3	θ31	θ32	1.0

Cosine similarity matrix

Figure 2. A matrix of cosine similarities for three vectors

The proposed method is summarized in following primary steps that are also shows in Figure 3.

Step 1: From the corpus (1,000 documents), a term_by_document matrix is created using only term counts. Before running the code, the following variables are set:

The stop list and the ignore characters are specified. We used the following as a stop list: { الكويت، الكويتية، الكويتي، الكويتيين، بالكويت، الانباء}. It has Kuwaiti words (since the corpus belongs to Kuwaiti newspaper) as well as the name of the newspaper.The ignore characters includ: {~, ` , ! , @ , # , £ , € , $, % , ° , ^ , & , * , (,) , - , _ , + , = , » , « , { , } , [,] , | , \ , / , : , ; , 0,1,2,3,4,5,6,7,8,9}.

The document frequency (DF) feature is set. We used a range of document frequency values to determine the optimal performance. The range is [2:10].Finally, the term_by_document matrix is weighted using TF-IDF weighting scheme.

Step 2: SVD is used to reduce thematrix generated in step1 to produce three matrices, U, S and V^T. The singular value (k) is set (choosing k is related to the corpus size. In our case, we propose using k=30). Hence, both terms and documents are represented by k-dimensional vectors in this vector space.

Step 3: A cosine similarities matrix is created using document vectors generated in step 2. Figure 3 shows how to form the cosine similarities matrix. The dimensions of this square matrix is the total number of documents in both dimensions (i.e rows and columns).

Step 4: SVD is used to reduce the cosine similarities matrix to create $V^{T'}$. That is, The SVD method is used one more time to truncate the the matrix generated in the previous step (step 3).

Step 5: A clustering technique (ME) is implemented on the $V^{T'}$ matrix. Figure 3 shows that the cosine similarity matrix $V^{T'}$ is used to cluster the documents vectors into a pre specified number (i.e. 10), which is the total number of categories in the used corpus.

Step 6: The performance is measured using the overall accuracy. Accuracy is simply the ratio of correctly predicted observations to the number of actuals. The WEKA [27] tool can give the accuracy since it has the option to previously assign the category for each document to be later used for measuring the accuracy.

Figure 3 summarizes the steps that include performing the LSI technique of the cosine similarities produced by the first LSI implementation.

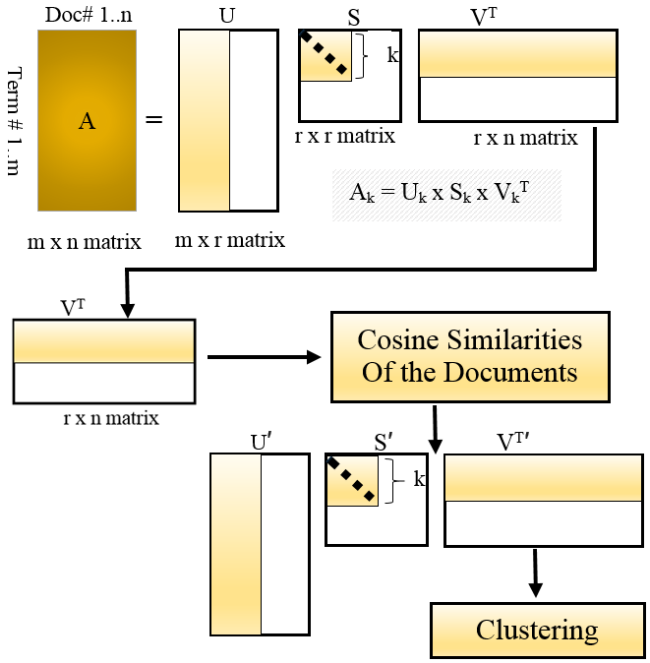


Figure 3. The steps of the proposed method

5. The Experimental Results

This section presents the experimentconducted to evaluate the effectiveness of the proposed method. We used an in-house Arabic corpus that contains 1,000 documents belonging to 10 different categories. The corpus contains more than 530,000 words that include more than 101,000 unique words. We got the documents from Alanba newspaper website in Kuwait [1]. The collected corpus contains different categories with focus on a specific topic, for examples, we collected 100 documents that belong to health with focus on the news stories related to blood diseases. In the Education category, we focused on the scholarship articles, and in the Technology, we focus on mobile technology, etc. Our corpus is not big enough; however,it gives an indication of the effectiveness of the proposed method. Table 2 shows the statistics of the corpus used.

Table 2. The corpus and the categories distribution

#	Category	Number of documents	Number of words	Unique words
1	Health	100	66,399	10,589
2	Economy	100	55,363	9,709
3	Sports	100	45,078	11,461
4	Islam and Sharia	100	89,822	14,583
5	Education	100	87,044	10,616
6	Arts and Artists	100	42,602	10,550
7	Municipality affairs	100	55,826	9,240
8	Tourism and travel	100	41,075	9,163
9	Security and law	100	34,644	9,611
10	Technology	100	22,031	6,097
	Total	1,000	539,884	101,619

We also specified the stop list and the ignore characters as described in the previous section. In the experiment, we ignored all English words and characters that may exist in the documents. That is, any English word or characterdiscarded and not considered in the LSI term_by_document matrix. We used Python programming language to generate the LSI term_by_document matrix. We used MATLAB programming languageto implement SVD and to create the cosine similarities matrix. The EM unsupervised clustering algorithm, available in WEKA, employed for clustering. Regarding DF feature, we selected different values of DF as shown in Table 3. The accuracies are also shown in Table 3.

Table 3. The accuracies of the proposed method

DF	LSI Accuracy (%)	Proposed Method Accuracy (%)
2	69.2	90.8
4	88.8	94.2
6	66.9	87.0
8	70.7	92.7
10	81.7	88.1
Average	75.46%	90.56%

The results in Table 3 indicates that the accuracy is increased in all DF values. In fact, we chose relatively small DF values as we have small corpus. The average enhancement of the overall accuracies is 15.1%.The information obtained in Table 3is presented in Figure 4.

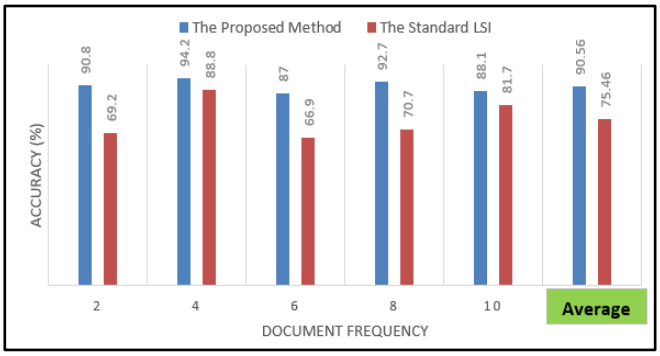


Figure 4. The performance of the proposed method

In our experiment, we used the singular values k=30. Intuitively, the clustering performance might be increased for other singular values (more or less). Nevertheless, we focused to present this novel method rather than investigating optimality of the singular values. However, our experimental investigation of text classification with cosine similarity shows that k=30 is the best value for our corpus. Even though, some previous studies showed that k=300-500 is reasonable value range, however, for the cases of millions of items.Reference [7] provided an empirical study of required dimensionality for large-scale LSI applications.

6. Conclusion and Future work

This paper demonstrates that LSI is a suitable technique for Arabic text clustering. We proposed a new method that integrate LSI and cosine similarity measure for the clustering task. The results show that

the hybrid method achieved a significant enhancement. For future work, we suggest to use this method to enhance the performance of IR applications. Specifically, the proposed method might be good for search engines and text categorization.

Acknowledgements

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.

References

[1] Alanba, "A Kuwaiti comprehensive newspaper," http://www.alanba.com.kw/newspaper/

[2] Al-Anzi, Fawaz S., and Dia AbuZeina. "Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing." *Journal of King Saud University-Computer and Information Sciences* (2016).

[3] Beil, Florian, Martin Ester, and Xiaowei Xu. "Frequent term-based text clustering." In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 436-442. ACM, 2002.

[4] Bellegarda, Jerome R., John W. Butzberger, Yen-Lu Chow, Noah B. Coccaro, and DevangNaik. "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, 1996.

[5] Bellegarda, Jerome R. "Exploiting latent semantic information in statistical language modeling." *Proceedings of the IEEE* 88, no. 8 (2000): 1279-1296.

[6] Bharti, KusumKumari, and Pramod Kumar Singh. "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering." *Expert Systems with Applications* 42, no. 6 (2015): 3105-3114.

[7] Bradford, Roger B. "An empirical study of required dimensionality for large-scale latent semantic indexing applications." In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 153-162. ACM, 2008.

[8] Dhillon, Inderjit S., and Dharmendra S. Modha. "Concept decompositions for large sparse text data using clustering." *Machine learning* 42, no. 1-2 (2001): 143-175.

[9] Elberrichi, Zakaria, AbdellatifRahmoun, and Mohamed Amine Bentaallah. "Using WordNet for Text Categorization." *Int. Arab J. Inf. Technol.* 5, no. 1 (2008): 16-24.

[10] Gomaa, Wael H., and Aly A. Fahmy. "A survey of text similarity approaches." *International Journal of Computer Applications* 68, no. 13 (2013).

[11] Google, "Translator," https://translate.google.com/

[12] Gotoh, Yoshihiko, and Steve Renals. "Document space models using latent semantic analysis." (1997).

[13] Homayouni, Ramin, Kevin Heinrich, Lai Wei, and Michael W. Berry. "Gene clustering by latent semantic indexing of MEDLINE abstracts."*Bioinformatics* 21, no. 1 (2005): 104-115.

[14] Kantardzic, Mehmed. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[15] Kuhn, Adrian, StéphaneDucasse, and Tudor Gírba. "Semantic clustering: Identifying topics in source code." *Information and Software Technology* 49, no. 3 (2007): 230-243.

[16] Lin, Tsau Young, and I-Jen Chiang. "A simplicial complex, a hypergraph, structure in the latent semantic space of document clustering." *International Journal of approximate reasoning* 40, no. 1 (2005): 55-80.

[17] Liu, Tao, Zheng Chen, Benyu Zhang, Wei-ying Ma, and Gongyi Wu. "Improving text classification using local latent semantic indexing." In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pp. 162-169. IEEE, 2004.

[18] Maletic, Jonathan I., and Naveen Valluri. "Automatic software clustering via latent semantic analysis." In *Automated Software Engineering, 1999. 14th IEEE International Conference on.*, pp. 251-254. IEEE, 1999.

[19] Osinski, Stanislaw, and Dawid Weiss. "A concept-driven algorithm for clustering search results." *IEEE Intelligent Systems* 20, no. 3 (2005): 48-54.

[20] Chattamvelli, Rajan. *Data mining algorithms*. Alpha science international, 2011.

[21] Sharifi, Beaux, Mark-Anthony Hutton, and Jugal K. Kalita. "Experiments in microblog summarization." In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pp. 49-56. IEEE, 2010.

[22] Sobh, Ibrahim, Nevin Darwish, and Magda Fayek. "A trainable Arabic Bayesian extractive generic text summarizer." In *Proceedings of the Sixth Conference on Language Engineering ESLEC*, pp. 49-154. 2006.

[23] Takçı, Hidayet, and Tunga Güngör. "A high performance centroid-based classification approach for language identification." *Pattern*

Recognition Letters 33, no. 16 (2012): 2077-2084.

[24] Tata, Sandeep, and Jignesh M. Patel. "Estimating the selectivity of tf-idf based cosine similarity predicates." *ACM Sigmod Record* 36, no. 2 (2007): 7-12.

[25] Theodoridis, S. and K. Koutroumbas, *Pattern Recognition*, Fourth Edition, Academic Press, 2008.

[26] Wang, Dingding, Tao Li, Shenghuo Zhu, and Chris Ding. "Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization." In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 307-314. ACM, 2008.

[27] WEKA, "Machine Learning Tool," http://www.cs.waikato.ac.nz/ml/weka/

[28] Xu, Changsheng, Yi-Fan Zhang, Guangyu Zhu, Yong Rui, Hanqing Lu, and Qingming Huang. "Using webcast text for semantic event detection in broadcast sports video." *IEEE Transactions on Multimedia* 10, no. 7 (2008): 1342-1355.

[29] Yeh, Jen-Yuan, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng. "Text summarization using a trainable summarizer and latent semantic analysis."*Information processing & management* 41, no. 1 (2005): 75-95.

[30] Yu, Bo, Zong-ben Xu, and Cheng-hua Li. "Latent semantic analysis for text categorization using neural network." *Knowledge-Based Systems* 21, no. 8 (2008): 900-904.