An Implementation of Local Sparsity Ratio-Mine Algorithm for Arabic Text Categorization

Sameer Nooh, and Nidal F. Shilbayeh

***Abstract*—**Because of the complexities of the Arabic language and the tremendous number of text documents existed in the internet, the development of Arabic text categorization systems is a challenging problem for researchers. In this paper, we developed a new Arabic text classification system based-on Local Sparsity Ratio Mine Algorithm (LSC-mine). The developed system is capable of detecting outlier points in a spatial space; the discovering process is accomplished through computing the Local sparsity ratio (LSC), which indicates the outlier-ness of a certain point. Several experiments have been conducted to ensure the success of the developed system. The system has been implemented and tested using datasets of different categories collected and gathered from online Arabic documents websites.

***Keywords*—** Text Categorization, Text Classification, LSC-mine, Text Clustering, Outlier Detection.

I. Introduction

With the tremendous increase in the spread of information on the internet, there is a need for a text classifier which can help in classifying text documents in order to facilitate retrieving relevant information [1]. In this process, there are some major issues that must be considered for text classification applications such as dealing with any unstructured contents in the data, defining the number of features, and selecting a suitable machine learning technique in order to deliver highly accurate context [2].

Text categorization (TC) is defined as “the process of classifying documents into predefined categories”. Another definition describes the categorization process as an assignment of category labels to natural language documents; each document may be included in more than one category [3].

Automatic TC is termed as an assignment process of every textual document into the most appropriate category, thereby; it is a learning process of classification schemes through employing training datasets [4-6]. However, automated techniques of TC paradigm encounter several problems that must be solved by any proposed technique.

The first problem occurs when a text categorization system treats documents as a repository of words, which makes it difficult to extract the most minimal and optimized set of features [7]. The second problem encompasses the fact that categorization techniques may have a biased behavior toward documents with specific features or characteristics [8]; the third problem focuses on information retrieval and data mining (DM) disciplines, which is the reality of dealing with unstructured data and may be the most essential issue [9].

In the training phase, the classified text documents are scanned to identify the index terms and the phrases for each category. Then, the testing phase assigns each unstructured text document into an appropriate category(s) depending on its contents. Finally, a validation test is used to corroborate any inconsistent judgment by the classifier [10], [11].

In this paper, our contributions can be summarized as follows:

- The developed system based-on the Local Sparsity LSC-mine algorithm [12-13]. The Proposed algorithm is an outlier detection algorithm that belongs to a clustering paradigm.
- The adopted algorithm is capable of detecting outlier points in a spatial space; the discovery process was accomplished through computing the local sparsity ratio (LSC), which indicates the outlier-ness of a certain point.
- The system has been implemented and tested using a dataset that consists of a collection of Arabic text documents gathered from internet websites.
- Several experiments have been conducted to ensure the success of the developed system.

This paper is organized as follows. Section II presents the necessary background information and related works. The Implemented text clustering algorithm is provided in Section III. In Section IV, and V we describe the implementation and the results of the analysis. Finally, we present the conclusion and propose future work in Section VI.

II. Literature Review and Related Works

An automatic TC concept has been proposed several times during the last 35 years and some surveys [14-16] managed to address the most common TC algorithms.

Between the 1960s and early 1980s, the published TC techniques were restricted so that all TC techniques classified documents using an expert system that manually inserted rules; this is known as the disjunctive normal form (DNF) [17]. The main limitation here was the need to predefine all rules for every category manually. However, since the 1990s machine learning concepts have been enhanced with TC algorithms development and new construction systems.

The TC problem is composed of several sub-problems, which have been studied intensively in the literature such as document indexing, weighting assignment document clustering, dimensionality reduction, threshold determination, and the type of classifiers.

Document indexing is associated with a method of extracting indexes from the document. There are two main

Sameer. Nooh. He is now with the Department of Computer Science, University of Tabuk, Umluj, Saudi Arabia (e-mail: snooh@ut.edu.sa).

Nidal. Shilbayeh. He is now with the Department of Computer Science, University of Tabuk, Umluj, Saudi Arabia (e-mail: nshilbayeh@ut.edu.sa).

approaches to accomplish document indexing; the first approach considers index terms as “bags of words” [18] and the second approach regards the index terms as phrases [19], [20]. A drawback of the first approach is that it complicates index term extraction through increasing the dimensionality of the document. The second approach extracts phrases either syntactically [17], statistically [19] or using a combined method between them [20]. Despite decreasing the dimensionality of documents and supporting higher semantic qualities; dealing with phrases reveals several challenges such as longer phrases that include more terms and consequently more synonyms, in addition; phrases have lower frequencies which may affect the weighting assignment process.

Classifying Arabic text documents requires preprocessing the documents by extracting the roots. This process is quite significant in terms of reducing the dimensionality of the documents. Several techniques have been developed to perform these preprocessing tasks such as stemming, root extraction and thesaurus comparison.

Root extraction is considered to be one of the most important steps during the preprocessing phase, it is quite important in terms of reducing the document dimensionality and increasing the accuracy of the categorization process. Several statistical approaches have been explored. One of the proposed techniques [21] applies a list-removing mechanism to extract irrelevant letters from an Arabic word. This technique tries to follow up unimportant characters through evaluating the word using a co-occurrence analysis formula.

Weight assignment techniques assigned a real number, ranging from 0 to 1 for all documents’ terms; weights will be required to classify newly indexed documents. Different information retrieval models use unique methodologies to compute these weights, for example; the Boolean model assigns either 0 or1 for each index term. In contrast, vector space model computes a tf-idf factor [18], which ranges from 0 to 1, this model is further described in the following section.

There are two categories for the learning-based TC algorithms; they are inductive learning algorithms and clustering-based algorithms. A TC algorithm uses different decision tree models to classify documents through building a tree by computing the entropy function of the selected index terms [7], [22] such as ID3 [19] and C4.5 [23], [24].

Another inductive learning algorithm based on probabilistic theory, such as one that emphasized naïve Bayesian models [25], [26] generated good results in the TC field. Historically, the most widely famous naïve Bayesian model was known as a binary independent classifier [27].

The hope in constructing TC algorithms that have the ability to learn, using the least amount of training, means that an efficient approach to classifying documents appropriately. The clustering issue is intensively emphasized in the science of statistics and will be discussed in detail in the following section.

One of the first attempts to address clustering techniques for the TC problem was stated by introducing a comparison between the k-means algorithms and hierarchical clustering algorithms [28]. The results showed better performance for the hierarchical algorithms although they were slower than the k-means algorithm.

Different TC methods have emerged to categorize documents, such as support vector machine and multi-class support vector machine SVM [23], [29], k-nearest neighbor KNN [29], least linear square fit LLSF [25], logistic regression LR [30] and ridge regression.

Linear SVM based algorithms use quadratic programming techniques to optimize the operation of structural risk maximization, which in turn tries to reduce any generalization error instead of reducing the empirical error [15], [29]. Various KNN models compute the distances between the document index terms and the known terms of each category by applying distance functions such as cosine, dice similarity or Euclidian or Minkowski functions, the returned classes are the kth classes with the highest scores.

An LLSF algorithm [31] measures the likelihood between the documents’ index terms and all existing categories using a linear parametric model, which consisted of three steps. In this model each document was associated with two vectors: I [dj] as an input weight vector of term T and O (dj) as an output vector of category C to which the document belongs.

The most delimited factor in applying an LLSF algorithm, though its results compete with an SVM algorithm, is its high space computational and space complexities. LR and RR algorithms calculate the likelihood between training set category’s index terms and document’s index terms using linear and binary regression models respectively.

III. The Modified Text Clustering System

In this paper, we developed a text clustering algorithm Based-on the Local Sparsity LSC-mine algorithm [12-13] and the well-known density-based outlier detection algorithm (local outlier factor) algorithm [32]. The algorithm calculates the distances between the nearest neighbors and the objects and assigns a degree of incoming data points. The assigned degree expresses the degree of isolation between these points and their neighbored points. Fig. 1 shows the flowchart of the Implemented algorithm.

IV. Implementation and Experimental Results

Our Implemented system has been demonstrated and tested on the following samples of Arabic documents:

Document sample 1:

“ سأل القوم عن اقتصاد الأردن و مدى قدرته على التأقلم في الظروف
الاقتصادية الراهنة، الأردن له اقتصاد قوي ويملك ثروات بشرية هائلة، الثروة
الحقيقية هي الإنسان."

Document sample 2:

"الإنسان العربي يعاني من مشاكل اقتصادية, تتمثل في البطالة, يجب حل
مشكلة البطالة من خلال استغلال الثروات في توظيف وإثراء الاقتصاد."

Step 1: The resulted documents after extracting the word’s roots:

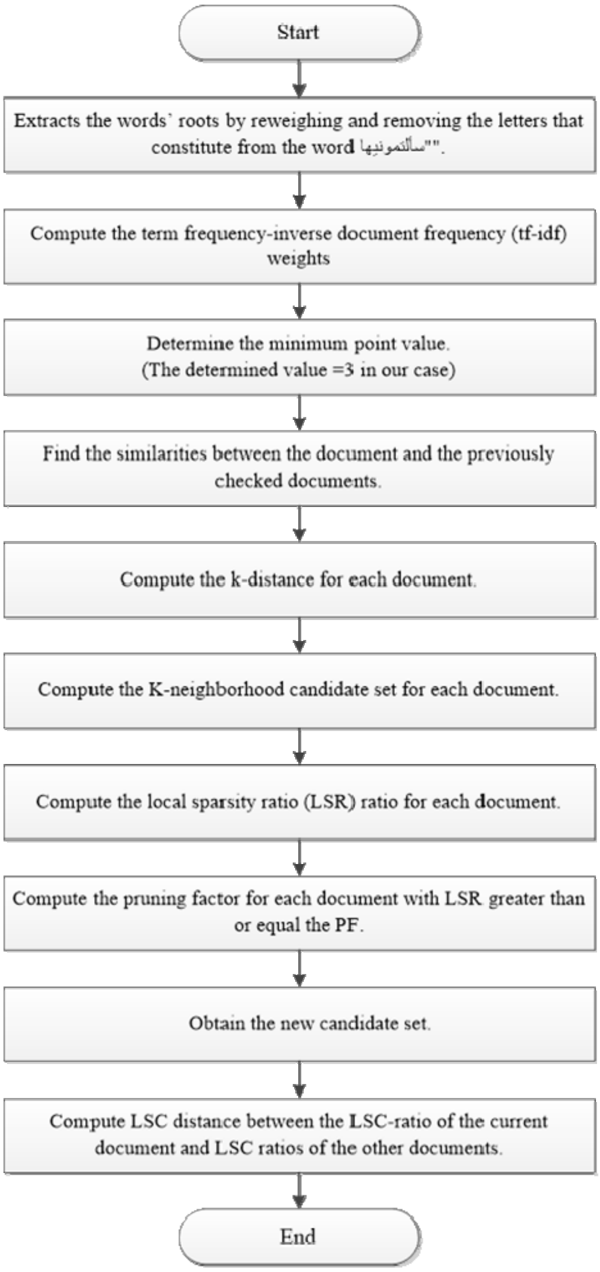
Start
Extracts the words' roots by reweighing and removing the letters that constitute from the word سألتمونيها"".
Compute the term frequency-inverse document frequency (tf-idf) weights
Determine the minimum point value.
(The determined value =3 in our case)
Find the similarities between the document and the previously checked documents.
Compute the k-distance for each document.
Compute the K-neighborhood candidate set for each document.
Compute the local sparsity ratio (LSR) ratio for each document.
Compute the pruning factor for each document with LSR greater than or equal the PF.
Obtain the new candidate set.
Compute LSC distance between the LSC-ratio of the current document and LSC ratios of the other documents.
End

Fig. 1 The Implemented algorithm.

Document sample 1:

“ سأل قام قصد ردن مدى قدر قلم ظرف قصد رهن ردن قصد قوى ثرى بشر هال ثرى حقق أنس”

Document sample 2:

“ انس عرب عانى مشكل قصد مثل بطل شكل بطل سغل ثرى وظف ثرى قصد”

Step 2: The calculated tf-idf.

The tf for the word قصد in document A is 3 / 3 = 1. Assuming that the word قصد has appeared in 12 documents out of 19 documents then the idf of the word قصد is Log (19 /12) = 1.079181246;

The weight of the word قصد in document A is 1* 1.079181246 = 1.079181246.

Step 3: Fig. 2 shows the resulted graph where nodes are labeled from A to S and the written numbers on the edges represent the similarities computed using the Cosine measurement formula.

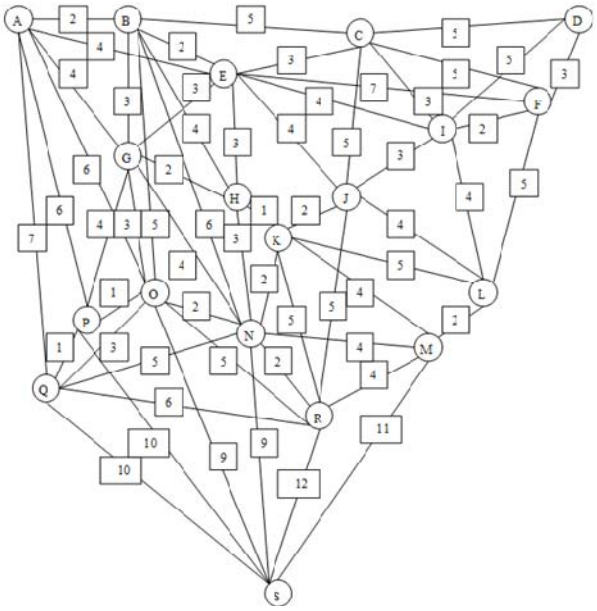

Fig. 2 The graph depicts the computed similarities in the documents

Steps 4 and 5: In this example, kdistance(A) is computed by looking at the most closest three points, which are {B, E, G}, the maximum distance between A and one of these points is equal to the kdistance of (A), all points that have less distance than the kdistance (A) are encompassed in $N_k(p)$) set, the cardinality N_k (*p*)| of this set is equal to the number of nodes; the cardinality |A| is equal to 3. Table 2 shows the computed kdistance (*p*), candidate set *p* (N_k (A)) and cardinality of A |N_k (A)|.

Step 6: The local sparsity ratio lsr_k(p) of an object p is computed using eq. 1 [12]:

$$lsr_k(p) = \frac{|N_k(p)|}{\sum_{o \in N_k(p)} distof N_k(p)} \qquad (1)$$

, where $|N_k(p)|$ is the cardinality of kdistance neighborhood of p and distofNk(P) consists of all the actual distances of objects in kdistance neighborhood of p. The numerator of the ratio is equal to card |*p*| and the denominator is equal to the sum of actual distances between A and points in N_k (*p*)). The computed result of lsr_k(A):

$$lsr_k(A) = \frac{card|A|}{sum\left(dist(A-B), dist(A,E), dist(A,G)\right)} = \frac{3}{10} = 0.3$$

Table 1. The documents and the similarities

Edge	Actual distance	Edge	Actual distance	Edge	Actual distance	Edge	Actual distance	Edge	Actual distance
A-B	2	C-D	5	F-I	2	J-R	5	N-S	9
A-E	4	C-E	3	F-L	5	K-L	5	O-P	1
A-G	4	C-F	5	G-H	2	K-M	4	O-Q	3
A-O	6	C-I	3	G-N	4	K-N	2	O-R	5
A-P	6	C-J	5	G-O	3	K-R	5	O-S	9
A-Q	7	D-F	3	G-P	4	L-M	2	P-Q	1
B-C	5	D-I	5	H-K	1	M-N	4	P-S	10
B-E	2	E-F	7	H-N	3	M-R	4	Q-R	6
B-G	3	E-G	3	I-J	3	M-S	11	Q-S	10
B-H	4	E-H	3	I-L	4	N-O	2	R-S	12
B-O	5	E-I	4	J-K	2	N-Q	5		
B-N	6	E-J	4	J-L	4	N-R	2		

Table 2. The computed kdistance (*p*), candidate set *p* (N_k (A)) and cardinality of A |N_k (A)|

label	Kdistance	Candidate set (N_k (p))	Cardinality |N_k (p)|
A	6	B, E, G, O, P	5
B	4	A, E, G, H	4
C	5	B, D, E, F, I, J	6
D	5	C, F, I	3
E	4	A, B, C, G, H, I, J	7
F	5	C, D, I, L	4
G	4	A, B, E, H, N, O, P	7
H	3	E, G, K, N	4
I	4	E, C, F, J, L	5
J	4	E, I, K, L	4
K	4	H, J, M, N	4
L	5	F, I, J, K, M	5
M	11	K, L, N, R, S	5
N	4	G, H, K, O, R	5
O	3	G, N, P, Q	4
P	6	A, G, O, Q	4
Q	5	P, O, N	3
R	5	J, K, M, N, O	5
S	11	M, N, O, P, Q	5

Step 7: The pruning factor (Pf) is the ratio of the sum of the absolute neighborhood distances to the overall sum of the actual neighborhood distances. The pruning factor PF (*p*) for point *p* is computed using the equation 2 [12]:

$$PF(A) = \frac{\sum |N_k(p)|}{\sum \sum_{o \in N_k(p)} dist of \, N_k(p)} \quad (2)$$

, where the numerator of the ratio is equal to the sum of cardinalities in N_K (*p*) and denominator is equal to the sum of the actual distances between p's neighborhood points and their neighborhoods' points. Once the pruning factor is obtained,

any object with a local sparsity ratio less than Pf is removed since it cannot be a potential outlier candidate. The pruning can remove more than half of the data thereby enhancing the performance of LSC-Mine.

The computed results of PF(A):

$$PF(A) = \frac{card|B| + card|E| + card|G|}{sum \; of \; dist\left((B - its \; neighbors), (E - its \; neighbors), (G - its \; neighbors),\right)} = \frac{12}{33} = 0.364$$

Steps 8: Once the PF is obtained, for every point p, if $Lsr_k(p) < PF(p)$ then *p* is removed since it cannot be a potential candidate. After examining all points in the space; the new candidate sets are created. In our example, G is removed from all candidate sets of other points because PF(G) < Lsr_k (G), therefore the new candidate set of A becomes {B, E} instead of {B, E, G}. It is noted that more than ten out of nineteen points have been removed.

Table 3 shows the computed LSR and PF values for all other points as well as the new candidate sets.

Step 9: The local sparsity coefficient of p denoted LSCk(p) is the average ratio of the local sparsity ratio of p to that of its k-nearest neighbors. The local sparsity coefficient ratio $LSC_k(p)$ is computed using the equation 3:

$$LSC_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lsr_k(o)}{lsr_k(p)}}{card|N_k(p)|} \quad (3)$$

, where $LSC_k(p)$ is the average ratio of the local sparsity ratio of p to that of its k-nearest neighbors. A high local sparsity coefficient indicates the neighborhood around an object is not crowded and hence a higher potential of being an outlier whereas a low local sparsity coefficient indicates a crowded neighborhood and hence a lower outlying potential.

In the example, the value is equal to the sum of all LSR ratios of p's neighbor points to the LSR ratio of p:

$$LSC_k(A) = \frac{\frac{lsr_k(B)}{lsr_k(A)} + \frac{lsr_k(E)}{lsr_k(A)}}{card|A|} = \frac{\frac{0.364}{0.3} + \frac{0.364}{0.3}}{2} = 1.21$$

Table 4 includes all LSC ratios for all candidate points. The three points S, R, and M is the highest degree of outlier-ness because they have the highest LSC ratios respectively.

Table 3. LSR, PF, and the new candidate sets

label	Lsr	PF	Marking flag	Old candidate set	New candidate set	Card.
A	0.227	0.333	R	B, E, G, O, P	B, E	2
B	0.364	0.297	N	A, E, G, H	E, H	2
C	0.231	0.303	R	B, D, E, F, I, J	B, E, F, I	4
D	0.23	0.263	R	C, F, I	F, I	2
E	0.304	0.291	N	A, B, C, G, H, I, J	B, H	2
F	0.267	0.253	N	C, D, I, L	I	1
G	0.304	0.333	R	A, B, E, H, N, O, P	B, E, H, O	4
H	0.444	0.338	N	E, G, K, N	E, K, N	3
I	0.313	0.268	N	E, C, F, J, L	F	1
J	0.308	0.309	R	E, I, K, L	E, I, K	3
K	0.444	0.3	N	H, J, M, N	H, M, N	3
L	0.25	0.278	R	F, I, J, K, M	I, M	2
M	0.25	0.214	N	K, L, N, R, S	K, N	2
N	0.385	0.338	N	G, H, K, O, R	K, O	2
O	0.444	0.257	N	G, N, P, Q	N, P	2
P	0.333	0.301	N	A, G, O, Q	O	1
Q	0.333	0.382	R	P, O, N	N, O, P	3
R	0.238	0.318	R	J, K, M, N, O	K, M, N	3
S	0.102	0.308	R	M, N, O, P, Q	M, N, O, P	4

Table 4. LSC ratios

Label	LSC ratio	Label	LSC ratio
A	1.4715	K	0.81
B	0.805	L	1.126
C	1.351	M	1.658
D	1.261	N	1.153
E	1.329	O	0.809
F	1.172	P	1.333
G	1.28	Q	1.163
H	0.85	R	1.511
I	0.853	S	3.461
J	1.148		

V. Implementation and Experimental Results

Several experiments have been conducted to ensure the eligibility of the developed algorithm. These experiments were accomplished using a dataset that consists of collection of Arabic-language; these documents are gathered from the internet websites.

Four different experiments were performed to detect the accuracy of the proposed algorithm; other experiments were applied to compute the time complexity of the developed code.

It is regarded that the implementation of the algorithm is done using the C#.NET 2012.

The implementation is comprised of three main phases: -

a. **The parsing phase:** This phase removes stop words, extracts roots of the words, and counts the number of words in the documents.
b. **The construction of document-term matrix phase:** this matrix is quite helpful in reducing the complexity of finding the word frequencies, which in turn are used to compute the tf-idf weights and thereby identify the document's keyword.
c. **The LSC-mine classification phase:** by applying the LSC-mine algorithm, which is used to classify arrived documents into their appropriate categories through computing the LSC ratio.

The dataset consists of 241 Arabic-language documents collected from the Internet sites; the dataset includes documents belonging to five categories; learning, computer, software engineering, and information systems and banks information. Experiments were repeated over various batches of documents, each of which has different sizes. The experiments were accomplished using 50, 100, 150 and 200 files. In every batch, the number of documents belonging to the same category is the same.

A. LSC-mine Accuracy Test

The accuracy of the developed classifier has been tested using 4 batches with different sizes 50, 100, 150, and 200. The accuracy is computed by dividing the number of truly classified documents over the number of total tested files. The experiment shows that the accuracy of the classifier is gradually increased as the size of the batch increases; this is due to the learning behavior that is played by the LSC-mine classifier. The accuracy of LSC-mine algorithm is equal to 0.6521 when the size of the batch is 200 documents. Fig. 3 shows the resulted accuracy test using different batches.

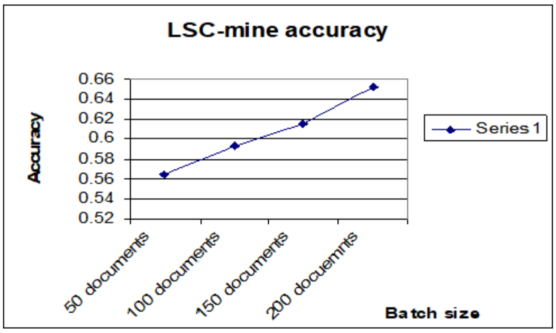

Fig. 3 The accuracy of the developed classifier using different batches

B. Time complexity Tests

Several experimental tests have been accomplished to analyze the time complexity of various phases of the LSC-mine classifier. Fig. 4 shows the resulted time complexity test of the parsing phase.

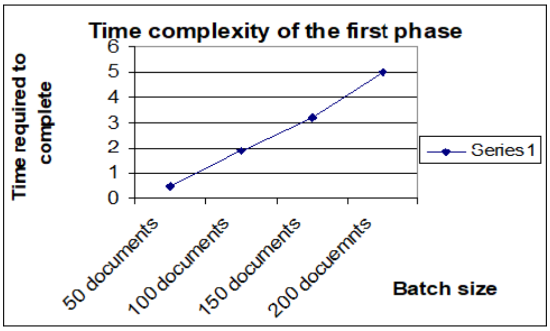

Fig.4 Time complexity of the Parsing Phase

The result shows that the parsing phase requires a time complexity that is linearly increased as the number of the size of batch increases. Fig. 5 shows the percentage between the time complexity and the batch size.

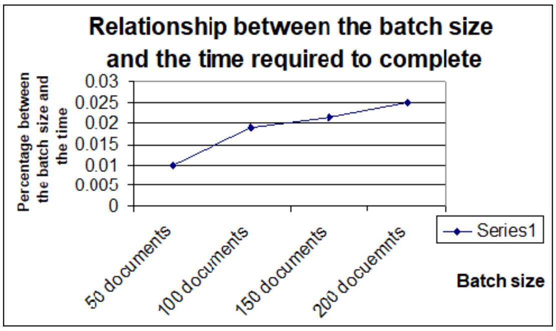

Fig. 5 The relationship between the time complexity and the batch sizes

The second phase is the Document-Term matrix construction phase (D-T matrix); this phase is considered as the shortest phase in terms of time complexity. Fig. 6 illustrates the time required to run different batches of files.

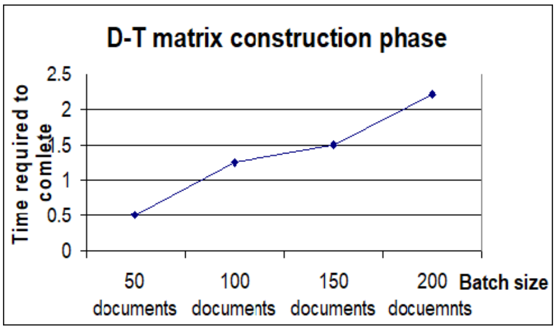

Fig.6 Time complexity of the D-T matrix construction phase

Fig.7 shows the D-T matrix constructions phase has a fluctuated curve; this is due to the fact that the construction phase is related to the number and the occurrences of words instances.

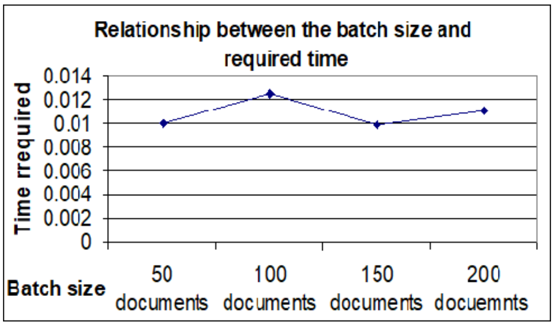

Fig. 7 Relationship between the batch size and the time complexity of the construction phase

The LSC-mine classification phase is regarded as the most crucial phase; it requires the computation of LSC-mine ratio. The most critical step in this phase is the necessity to compute the similarity between different documents of the corpus and the new document. Fig. 8 shows the time complexity of the LSC-mine phase.

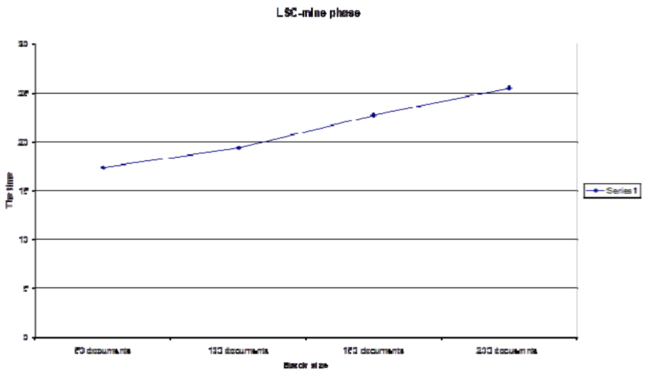

Fig. 8 Time complexity of the LSC-mine phase

Fig. 9 shows the Relationship between the batch size and the time complexity of the classification phase.

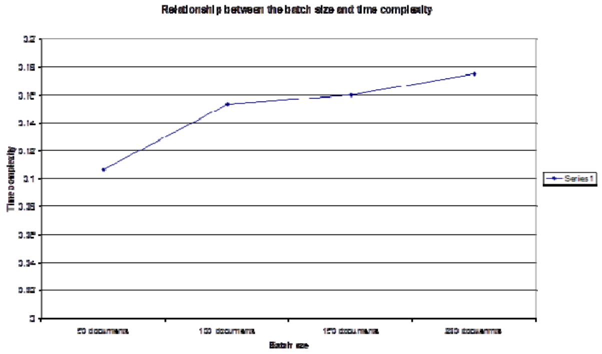

Fig. 9 Relationship between the batch size and the time complexity of the classification phase

Finally, the Total time complexity of the LSC-mine classifier is depicted in fig. 10. It shows that the classifier might be relatively inefficient in terms of time complexity as the size of the batch increases.

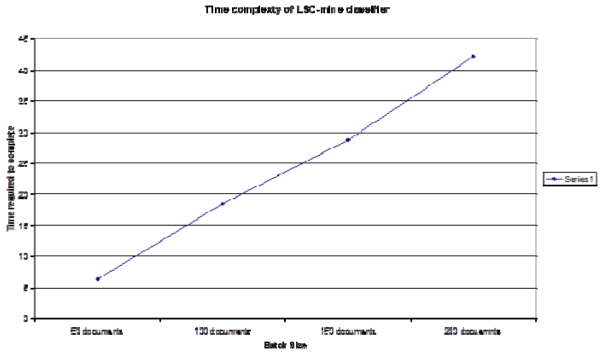

Fig. 10 The time complexity of the LSC-mine classifier

VI. Conclusions and Future Work

This paper aims at proposing a new text categorization technique to classify Arabic text documents. The proposed technique uses an LSC-mine algorithm [12-13], which is an outlier detection algorithm that computes the outlier-ness of a document from a certain category.

Experiments found that the accuracy of the implemented LSC-mine algorithm and the time required to complete classification of different document batches was performing better in comparison with other related algorithms implemented in Arabic text documents. These results show that the classifier completes other addressed algorithms, nevertheless; the algorithm requires improvements to reduce the time complexity especially reducing the time required to compute any similarity between a new document and previously stored documents. You may mention here granted financial support or acknowledge the help you got from others during your research work.

References

[1] G. Aggarwal, C.C. and Zhai, C. eds., 2012. Mining text data. Springer Science & Business Media.

[2] Pazzani, M.J. and Billsus, D., 2007. Content-based recommendation systems. In The adaptive web(pp. 325-341). Springer, Berlin, Heidelberg.

[3] Sebastiani, F., 2005. Text categorization. In Encyclopedia of Database Technologies and Applications (pp. 683-687). IGI Global.

[4] Dalal, M.K. and Zaveri, M.A., 2011. Automatic text classification: a technical review. International Journal of Computer Applications, 28(2), pp.37-40.

[5] Lam, W. and Lai, K.Y., 2001, September. A meta-learning approach for text categorization. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 303-309). ACM.

[6] Gao, S., Wu, W., Lee, C.H., Chua, T.S. and Chua, T.S., 2004, July. A MFoM learning approach to robust multiclass multi-label text categorization. In Proceedings of the twenty-first international conference on Machine learning (p. 42). ACM.

[7] Gabrilovich, E. and Markovitch, S., 2004, July. Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4. 5. In Proceedings of the twenty-first international conference on Machine learning (p. 41). ACM.

[8] Del Castillo, M.D. and Serrano, J.I., 2004. A multistrategy approach for digital text categorization from imbalanced documents. ACM SIGKDD Explorations Newsletter, 6(1), pp.70-79.

[9] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D., 2019. Text classification algorithms: A survey. Information, 10(4), p.150.

[10] Han, J., Pei, J. and Kamber, M., 2011. *Data mining: concepts and techniques*. Elsevier.

[11] Han, J., Kamber, M. and Pei, J., 2011. Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems*, pp.83-124.

[12] Agyemang, M., 2004. Algorithm for Mining Local Outliers. In Innovations Through Information Technology: 2004 Information Resources Management Association International Conference, New Orleans, Louisiana, USA, May 23-26, 2004 (Vol. 1, p. 5). IGI Global.

[13] Nooh, S, and Shilbayeh, N. 2019, Arabic Text Categorization based-on the Local Sparsity Ratio Mine Algorithm (LSC-mine), *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),* 8(5), pp.32-36.

[14] Schütze, H., Hull, D.A. and Pedersen, J.O., 1995. A comparison of classifiers and document representations for the routing problem. In Annual ACM conference on Research and Development in Information Retrieval-ACM SIGIR.

[15] Hmeidi, I., Al-Ayyoub, M., Abdulla, N.A., Almodawar, A.A., Abooraig, R. and Mahyoub, N.A., 2015. Automatic Arabic text categorization: A comprehensive comparative study. Journal of Information Science, 41(1), pp.114-124.

[16] Alghamdi, H.M. and Selamat, A., 2017. Arabic Web page clustering: A review. Journal of King Saud University-Computer and Information Sciences.

[17] Lewis, D.D., 1992, June. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 37-50). ACM.

[18] Salton, G. and Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), pp.513-523.

[19] Fuhr, N. and Buckley, C., 1991. A probabilistic learning approach for document indexing. ACM Transactions on Information Systems (TOIS), 9(3), pp.223-248.

[20] Yang, Y. and Chute, C.G., 1994. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems (TOIS)*, *12*(3), pp.252-277.

[21] Larkey, L.S., Ballesteros, L. and Connell, M.E., 2002, August. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 275-282). ACM.

[22] Breiman, L., Friedman, J.H. and Olshen, R.A., 2017. Classification and regression trees: Routledge.

[23] Joachims, T., 1998, April. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137-142). Springer, Berlin, Heidelberg.

[24] Cohen, W.W. and Hirsh, H., 1998, August. Joins that Generalize: Text Classification Using WHIRL. In *KDD* (pp. 169-173).

[25] Yang, Y. and Chute, C.G., 1994. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems (TOIS)*, *12*(3), pp.252-277.

[26] Li, Y.H. and Jain, A.K., 1998. Classification of text documents. *The Computer Journal*, *41*(8), pp.537-546.

[27] Robertson, S.E. and Jones, K.S., 1976. Relevance weighting of search terms. *Journal of the American Society for Information science*, *27*(3), pp.129-146.

[28] Jain, A.K., 2010. Data clustering: 50 years beyond K-means. *Pattern recognition letters*, *31*(8), pp.651-666.

[29] Johnson, D.E., Oles, F.J., Zhang, T. and Goetz, T., 2002. A decision-tree-based symbolic rule induction system for text categorization. *IBM Systems Journal*, *41*(3), pp.428-437.

[30] Zhang, J., Jin, R., Yang, Y. and Hauptmann, A.G., 2003, August. Modified logistic regression: An approximation to SVM and its applications in large-scale text categorization. In *ICML* (Vol. 3, pp. 888-895).

[31] Ciya, L., Shamim, A. and Paul, D., 2001. Feature preparation in text categorization. *Oracle Text Selected Papers and Presentations*, pp.1-8.

[32] Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J., 2000, May. LOF: identifying density-based local outliers. In *ACM sigmod record* (Vol. 29, No. 2, pp. 93-104). ACM

Sameer Nooh received the BSc. A degree in Computer Science from King Abdulaziz University, Jeddah, Saudi Arabia, and MSc Internet, Computer and System Security from University of Bradford, UK in Information Security in 2007. MSc consultancy from Liverpool John Moores University. Sameer finished his Ph.D. in Computer Science De Montfort University in Leicester, UK 2014. In 2015, Dr.Sameer joined the Computer Science Department, University of Tabuk, as an Assistant Professor in the Computer Science Department, University College, Umluj. His main areas of research interest are Information and System Security, Computer Science, and anything related to the Internet and computer. Since 2014 Dr. Sameer started some administrative assignments includes: Supervisor of Information Technology Unit, Vice-dean of University College, Umluj and now he is Dean of University College, Umluj, University of Tabuk, The northern area, Tabuk, Saudi Arabia

Nidal Shilbayeh received the BSc degree in computer science from Yarmouk University, Irbid, Jordan in 1988, the MS degree in computer science from Montclair State University, New Jersey, USA in 1992, and the PhD in computer science from Rajasthan University, Rajasthan, India in 1997. He is a Professor at the University of Tabuk. He was the Vice Dean at university of Tabuk, Saudi Arabia; He was the Vice Dean of Graduate Studies and Scientific Research at Middle East University, Amman, Jordan. He supervised many graduate students for the MS and PhD degrees. His research interests include Security (Biometrics, Identification, Privacy, Authentication, and Cryptography), Information Security (e-payment, e-voting, and e-government), Face Recognition, Digit Recognition, Watermarking, Embedding, Nose System, Neural Network, Image Processing, and Pattern Recognition.