International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 8, Issue 5, September - October 2019 ISSN 2278-6856

Arabic Text Categorization based-on the Local Sparsity Ratio Mine Algorithm (LSC-mine)

Sameer Nooh[1] and Nidal Shilbayeh[2]

[1]Department of Computer Science, University of Tabuk, Umluj, Saudi Arabia,

[2]Department of Computer Science, University of Tabuk, Umluj, Saudi Arabia,

Abstract: ***Outlier detection is an important research area in text mining, information retrieval, machine learning, and statistics as well as enhancing natural language processing paradigms due to the enormous numbers of new documents being utilized for various information retrieval systems. One of the most challenging problems in this context is addressing the text categorization problem with Arabic text documents. In this paper, we propose a new text categorization (TC) algorithm which classifies Arabic text documents using the local sparsity coefficient-mine algorithm (LSC-mine algorithm). The chosen algorithm is capable of detecting outlier points in a spatial space and clusters documents by computing the LSC ratio between the new document and the cluster's documents, which indicates the outlier-ness of a certain point. Several experiments have been conducted to ensure the success of the developed algorithm.***

Keywords: Text Categorization, LSC-mine, Arabic Language Text Clustering, Outlier Detection Algorithm.

1. 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], [3].

Manual text classification (TC) is the process of classifying text documents individually and this requires hiring someone who has good experience in categorizing documents. Nowadays, some of these systems accept between thousands and millions of documents every day, which create another tedious job for the classifier, who must have fully complete knowledge about all the topics in those documents.

Automatic TC is the method to classify an unstructured text document into the most appropriate category(s) depending on its contents using a machine learning technique [4]-[6]. However, automated systems using a text categorization paradigm encounter several problems that must be addressed 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].

Automatic TC process is classified as a DM branch due to the intensive deployment of learning schemes and mechanisms in its categorization process. Each DM system passes through training and testing phases to accomplish the categorization process.

In this paper, we aim to contribute to Arabic text documents categorization since it is one of the most challenging and complicated languages. Our contributions can be summarized as follows:

- Proposing a new TC algorithm to categorize Arabic language test files; the developed algorithm is based on classifying Arabic text documents using an LSC-mine algorithm [10];
- 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.

This paper is organized as follows. Section II presents the necessary background information and related works. The proposed text clustering algorithm is provided in Section III. In Section IV, Several experiments have been conducted to ensure the eligibility of the algorithm. Finally, we present the conclusion and future work in Section V..

2. LITERATURE REVIEW AND RELATED WORKS

An automatic TC concept has been proposed several times during the last 35 years and some surveys [11]-[13] managed to address the most common TC algorithms.

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.

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 [14] 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.

Another research has concerned [10], [11] with establishing a new model to extract the structure of Arabic noun phrases; such a model is quite useful to capture the semantics of the document's file. This model identifies two main categories to punctuate all noun phrases; eight different structures comprise these main categories. Despite its significance, the results of this work are not quite impressive.

Defining a formal descriptive model [15] to refine the syntax of Arabic sentences; the benefits of constructing such a model is in identifying the semantics of Arabic language sentences. The model works through exploring the grammatical aspects of a sentence using the Definite Clause Grammar; this scheme is quite useful in exploring non-terminal arguments such as the gender, number and person agreements. The exploration process is accomplished through studying the context similarity of the sentence.

Weight assignment techniques assigned a real number, ranging from 0 to 1 for all documents' terms [16]; 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 [17], 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], [18] such as ID3 [19] and C4.5 [20], [21].

Another inductive learning algorithm based on probabilistic theory, such as one that emphasized naïve Bayesian models [22], [23] generated good results in the TC field. Historically, the most widely famous naïve Bayesian model was known as a binary independent classifier [24].

- 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.

3 The Proposed Arabic Clustering Algorithm

In this paper, we propose a text clustering algorithm based-on the Local Sparsity LSC-mine algorithm [10]. LSC-mine algorithm is an extension of well-known density-based outlier detection algorithm called Local outlier factor (LOF) algorithm [15]. LSC-mine algorithm computes the distance of an object and those of the nearest neighbors without actually computing their reachability distances and local reachability densities. The main goal of LSC-mine algorithm is to assign a degree of incoming data points, which expresses the degree of isolation between these points and their neighbored points.

This paper contributes a new Arabic text clustering algorithm based on LSC ratio to classify incoming documents. The proposed algorithm follows the following sequence of steps:

Do the following for each new document:

1. The preprocessing phase: extracts the words' roots by reweighing and removing the letters that constitute from the word ""سألتمونيها"". This step is used to avoid considering words like طلاب ,طالب ,طلبas independent words.
2. The weighting assignment phase; compute the term frequency-inverse document frequency (tf-idf) weights [17] to obtain the single-word index terms for the document along with their weights, which are considered as the vectors of each document. Then determine the minimum point value, which sets the minimum number of neighbored documents or minimum number of documents that will not make the incoming document an outlier. The determined threshold is equal to 3.
3. Find the similarities between the document and the previously checked documents.
4. Compute the k-distance for each document.
5. Compute the K-neighborhood candidate set for each document.

6. Compute the local sparsity ratio (LSR) ratio for each document.
7. Compute the pruning factor for each document with LSR greater than or equal the PF.
8. Obtain the new candidate set.
9. Compute the LSC ratio using the new candidate set.
10. Compute LSC distance between the LSC-ratio of the current document and LSC ratios of the other documents.
11. The document that forms the least LSC distance with the new document indicates the candidate set that will contain this document.

The root extraction step can be done by executing the following steps:
1. Read the words from the document.
2. If the word consists of less than 3 words then remove the word from the document and consider it as a stop word.
3. If the word letters matched with any litters of the word ""سألتمونيها" then insert the letters in a temporary array named Temp_Array; Otherwise words are stored in array called Root_Array.
4. If the letter is ئ and is followed ا then insert the letter in the Root_Array as يletter.
5. If n < 3 then do step 6
Else do step 7
6. Root extraction process for third-based verbs
Do while Number of elements (n < 3)
a. If first letter is م and the last letter is ن then store the letter ن in the Root_Array.
b. If the number of elements of the Temp_Array is more than two AND the last letter in the word is ن, and the second last element is one of the following characters (أ, و, ي); store the third last word of the Temp_array in the Root_Array.
c. If the number of elements of the Temp_Array is more than two elements, and the last element of that array is أ and the second last element of the array is وAND the last letter of the original word is أ then store the third last word of the Temp_array in the Root_Array.
d. If the number of element of the Temp_Array is more than three AND the last letter of that array is Temp_Array is ت or ةAND the second last letter of that array is اAND the third last letter is يthen store the fourth last letter of the Temp_Array in the Root_Array.
e. If number of the word's letters in the Temp_Array is more than two letters and the last letter was ة or هand the second last letter is يand store the third last element of the Temp_Array in the Root_Array.
f. If the number of elements of the Temp_Array is more than one element and that element is one of the following characters ي, و, أ, ت, ه, ة) and the letter is the last character in the original word then remove that letter from the Temp_Array.
g. If the number of elements of the Temp_Array is more than one element and that element is one of the following characters ى, ي, و, أ) and the letter is the last character in the original word then store the second last letter existed in the Temp_Array in the Root_Array.
h. If the number of elements of the Temp_Array is more than two characters AND the last letter of that array isت AND the second last letter is اand the letter تis the letter of the original word. Store the third last letter in the Root_Array.
i. Else store the last letter of the Temp_Array in the Root_Array.
j. Arrange the letters of the Root_Array as appeared in the original word.
7. Root extraction process for fourth-based verbs
Do while Number of elements (n < 4)
a. If the word's letters is equal to five or six AND first letter is نthen store the letter نin the Root_Array.
b. If the number of elements of the Temp_Array is more than two AND the last letter in the word is ن, and the second last element is one of the following characters (أ, و, ي); store the third last word of the Temp_array in the Root_Array.
c. If the number of elements of the Temp_Array is more than two elements, and the last element of that array is أand the second last element of the array is وAND the last letter of the original word is أ then store the third last word of the Temp_array in the Root_Array.
d. If the number of elements of the Temp_Array is more than one element and that element is one of the following characters (ى, ي, و, أ) then store the second last letter existed in the Temp_Array in the Root_Array.
e. If number of the word's letters in the Temp_Array is more than two letters and the last letter was ة or هand the second last letter is ي. Store the third last element of the Temp_Array in the Root_Array.
f. If the number of elements of the Temp_Array is more than one element and that element is one of the following characters ن, ه, ة)) then remove that letter from the Temp_Array.
g. If the number of elements of the Temp_Array is more than two characters AND the last letter of that array is ت AND the second last letter is اand the letter تis the letter of the original word. Store the third last letter in the Root_Array.
h. Else store the last letter of the Temp_Array in the Root_Array.
i. Arrange the letters of the Root_Array as appeared in the original word.

4 EXPERIMENTS AND EVALUATION

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

The implementation is comprised of three main phases: -
1. **The parsing phase:** This phase removes stop words, extracts roots of the words, and counts the number of words in the documents.

2. **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.
3. **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. 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.

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. Figure 1shows the resulted accuracy test using different batches.

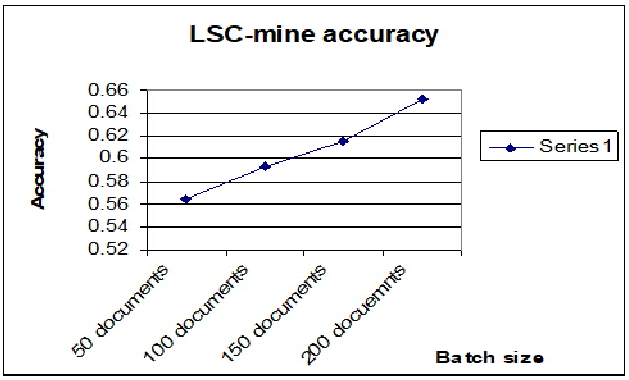

Figure 1 The accuracy of the developed classifier using different batches

5. CONCLUSION

Classifying Arabic text documents has not advanced for several years due to the difficulty in dealing with the Arabic language and a shortage of funding researches related to this topic. This paper aims at proposing a new text categorization technique to classify Arabic text documents. The proposed technique uses an LSC-mine algorithm [10], 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 LSC-mine algorithm is equal to 65.12%. The proposed algorithm contributes to the promotion of research on the classification of Arabic text documents

References

[1] 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] 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.
[11] 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.
[12] 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.

[13] Alghamdi, H.M. and Selamat, A., 2017. Arabic Web page clustering: A review. Journal of King Saud University-Computer and Information Sciences.
[14] Larkey, Leah S., Lisa Ballesteros, and Margaret E. Connell. "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, 2002.
[15] 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.
[16] Sebastiani, F., 2002. Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), pp.1-47.
[17] Salton, G. and Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), pp.513-523.
[18] Breiman, L., 2017. Classification and regression trees. Routledge.
[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] 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.
[21] Cohen, W.W. and Hirsh, H., 1998, August. Joins that Generalize: Text Classification Using WHIRL. In KDD (pp. 169-173).
[22] 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.
[23] Li, Y.H. and Jain, A.K., 1998. Classification of text documents. The Computer Journal, 41(8), pp.537-546.
[24] 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.

AUTHOR

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.