Classification of Arabic Documents depending on Maximal Frequent Itemsets

Zeyad Hamid1, Prof. Dr. Hussein K. Khafaji2
1College of Information Technology, University OF Babylon, Iraq
2Computer Communication Engineering Dept, Al-Rafidain University College, Iraq
Zeyadhamid2@gmail.com
dr.hkml811@yahoo.com

Abstract. In this paper we introduced techniques for classifying Arabic documents depending on association rules built from maximal frequent itemsets. Parallel Maximal Itemset Miner Algorithm (PMIMA) adopted several conditions to prune search space parallelly introduced for extracting maximal frequent itemsets. Rule length, rule weight and rule majority are three classification methods exploited to classification Arabic documents. Comparing with classification results obtained depending on all frequent itemsets extracted by Apriori, we proved efficiency of ours approach.

1. Introduction
Due to unexpected increasing of textual data amount that available electronically, it is become necessary to develop an automatic way that reduces required time and effort for extracting useful information. Data mining involves diverse range of techniques that can be used for handling and getting beneficial hidden information, these including for instance clustering, summarization, classification [1].

Documents classification is important text mining task regarding to assign a predefine class name to document depending on the document’s features [2]. It is easy to understanding and classifying a large number of documents by the human but takes long time and effort while this process needs short time as soon as we apply machine learning and data mining techniques. Documents classification is interdisciplinary research field including three main topics which are Natural languages processing (NLP), text mining and machine learning. Nowadays documents classification required in many domains like e-mail spam filtering, classify online articles and information retrieval.

Arabic language is the primary language in all Arabic countries that used by more than 422 million persons [3]. In addition, it considers the fifth spoken language around the word [4]. Therefore it is important developing electronic ways for classifying Arabic text in order to facility retrieving information concerning to this widespread language.

Since Association rules has discovered by Agrawal et al. in 1993, it used for solving many problems. First of all it exploited for extracting correlations that existing among basket transactions to study behavior of customers purchasing [5]. Then association rules mining utilizing for solving many problems like text classification. Building a textual classifier depending on association rule mining involves two main stages generating frequent itemsets and building classification association rules. Generating frequent itemsets considers the most challenging stage in this process because of volume of features in textual data. Therefore, sequential and parallel approaches were used to generate frequent itemsets[6]. Sequential approaches presented logical methods for enumeration and generating frequent
itemsets like Apriori[7], FP growth[8] and sampling [9]. On other hand parallel approaches attempted parallelizing extracting frequent itemsets process by partitioning candidate itemsets as well as database among multiple processors like data distribution [10], parallel Association rules [11] and hybrid distribution [12]. In spite of several sequential and parallel techniques have been proposed, finding frequent itemsets still expensive operation in term of time execution and storage space requirements. Therefore, maximal frequent itemsets considers alternative approach to generating all frequent itemsets for decreasing time and storage space requirements. In this research we construct an Arabic documents classifier through introduced a new parallel algorithm for generating maximal frequent itemsets.

2. Related Works

Many of methods have been used for classifying both English and Arabic documents. Joachims T (1998) used support vector machine (SVM) for text classification. The proposed method overcome other mechanisms by extracting features automatically [13]. Do to important of preprocessing stage and features chosen on the accuracy, scalability and efficiency of text classifier, Jingnian Chen et al (2009) established their paper. Multi-Class Ratio (MCR) and class discriminating measure (CDM) features evaluation metrics were used with Naive Bayes classifier [14]. Vishwanath Bijalwan et al (2014) applied k-Nearest Neighbors (KNN) to assort documents. The obtained result was better than Naive Bayes results dispite of time complexity [15]. Scientific papers classified by modified form of TF-IDF in Arlina D’cunha et al (2015) search paper according on features and structure of the processed document [16]. Lucas Dixon et al (2018) explained resulted bias of classification process can be happened because imbalanced training dataset [17].

Regarding Arabic document classification, association rule mining technique used by Al-Radaideh et al (2011). They applied ordered decision list, weighted rules and majority voting classification methods. Majority voting overcomes other two methods in terms of accuracy of document classification [18]. Cherif et al (2015) conducted a statistical analysis on tweeter datasets to determine author’s viewpoint about a particular theme [19]. Tarek Kanan (2015) introduced a P-stemmer for stemming Arabic words. In order to ptove effectiveness of proposed stemmer, NB, SVM, and RF used and showed better classification results than five Larkey stemmers. In addition SVM achieved highest accuracy comparing with others classifiers [20]. Ayedh et al (2016) discussed impact of pre-processing steps of Arabic text on the accuracy of SVM, NB and KNN classification techniques. Through their paper, they proved right selecting of pre-processing steps will improve the classification process. As well as the experiments showed SVM achieved 96.74% micro-F1 by combination normalization and stemming as pre-processing steps [21]. Al-Anzi et al (2017) exploited singular value decomposition for obtaining textual features according to latent semantic indexing. Then, they used cosine similarity measure for classifying Arabic text. Their approach for extracting features yields better results than TF-IDF technique. In addition KNN and SVM depend on cosine measure overcome six others classifiers [22]. Origin-stemmer proposed by Said et al (2017) to point out some problems of Khoja's stemmer. They used Chi-square technique for features selection and decision tree for classification [23]. Al-Radaideh et al (2018) decreased number of features by using term weighting and reduction idea of rough set. They introduced multiple minimal reduce extraction algorithm which outperformed single reduction in classification accuracy with 94%, meanwhile single reduct recorded 86% [24]. By using improved Chi-square technique Bahassine et al (2018) decreased the used features in the classification operation. They developed M matrix for finding relationships among attributes. Proposed method with SVM gave better classification results than traditional Chi-square, mutual information and information gain techniques [25].

3. Association Rules Mining

Association rule mining is one of the important data mining techniques exploited for finding unobserved correlations among itemsets in a database. This means existing specific item reflects present others items with specified probability. To generate a desired association rules, two measures have been used confidence and support. Support of rules is the number of the transactions that contain items to the total
number of transactions in database, while confidence is the ratio of present particular items in case of existing others items in the same transactions [7].

The formal definition of association rules: Let \( I = \{i_1, i_2, i_3, i_4, \ldots, i_m\} \) be a set of items. Let \( D \) be a set of transactions, where each transaction \( t \) is a set of items such that \( t \subseteq I \). Each transaction has a unique identifier, called TID. A transaction \( t \) contains \( X \), a set of some items in \( I \), if \( X \subseteq t \). An association rule is an implication of the form \( X \Rightarrow Y \), where \( X \subset I \), \( Y \subset I \), and \( X \cap Y = \emptyset \). The rule \( X \Rightarrow Y \) holds in \( D \) with confidence \( C \) if the segment of transactions that also contain \( Y \) in those which contain \( X \) in \( D \) is \( C \). The rule \( X \Rightarrow Y \) has support \( S \) in \( D \) if the segment of transactions in \( D \) that contains \( X \cup Y \) is \( S \). Given a set of transactions \( D \), the aim of mining association rules is to generate all association rules, which have support and confidence equal and larger than the minimum support (\( \text{minsup} \)) and minimum confidence (\( \text{minconf} \)) [7].

4. Maximal Frequent Itemsets
As mentioned before mining association rules process involves two main isolated stages extracting all frequent itemsets and extracting interesting association rules. First stage consumes most of the time comparing with the second step especially when set up a low minimum support and that lead to increased storage spaced required for processing itemsets[4]. So, many algorithms have been suggested for reducing computational time and storage space that required for discovering all frequent itemsets. Unfortunately, traditional algorithms generating a big number of frequent itemsets especially when set a low minimum support value and there is a dense database. So, to accelerate extracting itemsets process and reducing required computational time, researchers proposed a maximal frequent itemsets approach as an alternative of finding all frequent itemsets. Maximal frequent itemsets is suggested for removing repeated itemsets can be generating when using traditional algorithms. All superset of maximal frequent itemset are infrequent while all it subset are frequent. In other word maximal frequent itemsets represent a cover of its subset so can be used for driving all other frequent itemsets[26]. Maximal frequent itemsets involve compressed and lossless information of all frequent itemsets. It compressed because it contains a decreased and precise itemsets than all frequent itemsets giving an opportunity for extracting more interesting itemsets by reducing minimum support value [27].

5. Multi-Thread Process
Central Processing Unit (CPU) can consist of single core or Multi-cores. Multi-thread is the process of dividing the program into several tasks to execute them in single core or multi-core environment in order to increase speed of the program execution. Multithreading process shares processor time through uses switching technique to exchange processor among various threads. Threads in multithreading approach shares resources of the system which differ from multiprocessing when each process have its own resources of the system. Both multiprocessing and multithreading are techniques have been used to exploited system resources and for increasing speed of program execution time. Mining all frequent itemsets is a consuming time process especially in dense database [10]. So, our proposed method used threading principles with queue data structure to speed up generating maximal frequent itemsets of textual Arabic.

6. Proposed Techniques for Classification Arabic Documents
In this research, a machine learning algorithm based on association rules is presented as shown in figure 1. The proposed algorithms are dedicated for Arabic text classification. These algorithms are for training and testing phases. The training phase involves extracting maximal frequent itemsets in training dataset and classification rules extraction to build a classifier. While in testing phase we performed experiments to measure the accuracy of classification process.
7. Generating Maximall Frequent Itemsets
Maximal frequent itemsets extracting process involves series of steps as shown in figure 2.

**Figure 1.** Proposed technique
Figure 2. generation of maximal frequent itemsets

7.1. Pre-processing steps
In order to get accurate features (wordsets) from textual documents and for reducing documents size, pre-processing step is necessary. This step involves tokenization, stop word removing and stemming. Tokenization splits the documents into separated words. Then stop words remove from the documents. Words not reflect the document topic consider ineffective features of the documents like (is, the, by) in English language and ( ﻋﻠﻰ, ﺑﻲ) in Arabic language. By removing stop words we can obtain reduced volume of document. In Arabic and others languages many words derived from one root word for example in English language (written, writing, wrote) belong to write word which are correspond ( مكتوب, كتابة, كتاب ) in Arabic language so, we used stemming to remove prefixes, suffixes and infixes from these words and return them to their roots. In this research we exploited ISRIstemmer [25] to extract root of each word that obtained from stop word removing step. Words generated from stemming step are weighted using term frequency inverse document frequency (TF/IDF) to point out importance of each word in the document. TF is how many words (w) repeated in document (d) while number of repetitions of word in all documents represent (IDF) [26]. IDF of word (w) in a document (d) can calculated as shown in equation (1).

\[
\text{IDF}_{w,d} = \log \left( \frac{N}{n_w} \right) \quad \text{...(1)}
\]

Where:
N: Total number of documents
nw: number of documents contain word w

Word weight can be obtained through using equation (2)

\[
W_{w,d} = \text{TF}_{w,d} \times \text{IDF}_{w,d} \quad \text{...(2)}
\]

Where:
TF \( w,d \): (w) word frequency in document (d)
IDF \( w,d \): Inverse documents frequency of word (w) in document (d)
7.2. Initial Border Generation Algorithm

Performance of PMIMA algorithm is highly depending on the selected initial border, so it is important to pick out the best level in the search space to become an initiation point for the searching operation. In our algorithm we applied an equation allowing generating initial border that lay in middle place between the first level and last level that may contain maximal frequent itemsets. This equation is equivalent to support value of first element plus support value of last element in the first level dividing by general minimum support that determining by user as shown in equation (3).

\[
\text{InitialBorder} = \frac{\text{support value of first element in FL} + \text{support value of last element in FL}}{\text{general minimum support}} \ldots \ldots 3
\]

After determining the frequent item of first level and initial border value we then extracting the itemsets of initial border through using a special algorithm proposed by authors to achieve this goal. The algorithm depending on a moving several windows generating from frequent itemsets in the first level over others member of frequent itemsets in same level. It is written in a self-documentation manner and it guarantees that the generated itemsets are sorted if the frequent itemsets of first level are presented to algorithm in sorted form. If the number of frequent itemsets in the first level equal N and initial mining border equal to K, then we have the following equations which are very useful to design Initial border generator algorithm to mine K-itemsets in the initial border as shown in figure (3).

\[
\text{Window} = k-1 \ldots \ldots (5)
\]

\[
\text{Moves for each window} = N-Lk \ldots \ldots (6)
\]

Where Lk is sequential number of the last item in window in the frequent itemset of first level. So the total moves over first level of frequent itemsets to generate itemsets of initial border is equal to total moves for each window. The initial border that generated in the previous step will be divide according to the joining step fact of Apriori algorithm where two itemsets can be joined if their element identical except last element of both [1].

![Algorithm: Initial border Generator Algorithm](image)

**Figure 3.** Initial border generator algorithm
7.3. Parallel Maximal Itemset Miner Algorithm (PMIMA)

To paralyze maximal frequent itemsets extraction process we create a number of threads equal to the number of queues that obtained from previous step and assign each queue to one thread as shown in figure (4). The thread in parallel maximal itemset algorithm will take the itemset in the queue and classifying it into maximal, frequent and infrequent itemset. If itemset is maximal will keep it in the queue. If the itemset is frequent the algorithm will generate all superset for that itemset and store them in the queue, while if the itemset is infrequent the algorithm will generate all subset of that itemset and store them in queue. All these types of itemset will submitted to some conditions that used to decide whether we should put the supersets or subsets in the queue in order to reduce the search space. The first condition we can used to avoid adding the itemsets and consequently all subset is if \( Z=\{x_1, \ldots, x_{k-1}, y_{k-1}\} \) is an infrequent \( k \)-itemset that can be generated from two \( (k-1) \)-itemsets; \( X=\{x_1, \ldots, x_{k-1}\} \) and \( Y=\{x_1, \ldots, y_{k-1}\} \) such that \( X \) is frequent and \( Y \) is infrequent or vice versa, then \( Z \) has no sub itemset, \( W \), that is maximal itemset. So we can remove \( Z \) itemset from the queue and ignore all its subset and its superset. The second condition we can exploited to accelerate extracting maximal frequent itemsets if \( Z \) is an infrequent \( k \)-itemset and all its entire sub\((k-1)\) -itemsets are infrequent then it may have sub itemsets that are maximal itemsets in another level of the lattice. So, all the generated infrequent itemset adding to the queue and ignore the original infrequent itemset. Another condition used in this algorithm if \( Z \) is an infrequent \( k \)-itemset, then \( Z \) can be generated from infrequent and maximal frequent itemset. So if we face this type of itemset we need just check whether the frequent itemset subsets of that itemset is maximal or not so we need to add these frequent itemsets to queue and ignore their parent and all its superset. This process continues until all itemsets in each queue are maximal.

Figure 4. Parallel Maximal Itemsets Miner Algorithm
8. Association Rule Extraction
Maximal frequent itemsets extracted from documents in previous step utilize for constructing associative classifier rules. The rule form in proposed system composes of one or several terms (words), in the left hand side and class label on the right hand side as the following formula 7:

\[(\text{Word}_n \rightarrow D_c) \quad \text{conf:sup} \quad \ldots \quad (7)\]

Where \(\text{Word}_n\) is a series of maximal frequent itemsets expresses document in training dataset and \(D_c\) represents document class associated with this document. In order to specify importance of the rule we need compute two measures confidence and support of the rules. Support of rules represents the number of documents in training dataset that contains both \(\text{Word}_n\) and \(D_c\) as shown in formula 8.

\[
\text{Support}(\text{Word}_n \rightarrow D_c) = \frac{\text{Word}_n}{\text{Number of documents in training dataset}} \quad \ldots \quad (8)\]

Confidence of rules represents the number of documents that contain both \(\text{Word}_n\) and \(D_c\) to the total number of documents that contain \(\text{Word}_n\) in dataset as shown in formula 9.

\[
(\text{Word}_n \to D_c) = \frac{\text{Word}_n \cup D_c}{\text{support}(X)} \quad \ldots \quad (9)\]

Figure 5 shows Classification Rules Extracting algorithm (CRE) for generating classification rules in proposed system.

Classification will submit to pruning process according to predefined confidence and support values. Some of rules not possess classified power for recognizing documents so by removing these rules depending on specified confidence and support values, the number of rules that form the classifier will reduced and get more accurate classification results.

9. Classification Phase
To ensure of efficiency of built associative classifier, three methods were used for classification testing documents rules length, class majority and rules weight. In first technique of classification, the tested document will be classified according to longest rule in classifier. In other words, rule have longest body that matches document will be used for predicting by assign rule head to this document as shown in figure 6.
Class majority classification methods finds all rules covered by tested document, if all generated rules have identical head, the rule head will be attached to test document, otherwise extracting the most bodies of rules covered by the tested document and assign rule's head to that document as shown in figure 7.

![Figure 7. Class majority classification technique](image)

Figure 7. Class majority classification technique

Document under consideration will be classified using rule weight technique depending on calculating weights which are confidence and support of rule. Firstly, if all rules bodies correspond with tested document and have same class, then labelling this document with this class, otherwise, rules that have the same class gathering depending on their classes. For each class, calculate confidence and support weights. The confidence of class computed by summing all rules' confidence under this class divided by general predefined confidence value. While support of class equal to summing all rules' support under specific class divided by general predefined support value. Tested document attached with the class have highest confidence and support values as shown in figure 8.

![Figure 8. rule weight classification technique](image)

Figure 8. rule weight classification technique

10. Experiments and Evaluation
Comparison between PMIMA and Apriori algorithms regarding extraction frequent itemsets, generation association rules and accuracy of document classification presented in this section. Accuracy of documents classification explains the number of correctly classified documents among all documents in the test document collection as illustrated in formula 4.
Where $TC$ represents the number of correctly classified documents and $N$ is the total number of tested documents.

### 10.1. Dataset Description

To prove efficiency of proposed system, 1200 of Arabic documents involves sport, economic, law and astronomy categories taken from [28] was used for training system and extracting maximal frequent itemsets as well as generating association rules while 320 others documents dividing equally among categories i.e. 80 for each category exploited for classification phase.

### 10.2. Frequent Itemsets And Association Rules Generation

PMIMA executed with 40% to 80% minimum support thresholds and generated results compared with Apriori results in terms of number of extracted itemsets and execution time as appeared in table (1). It is obvious the deference between the number of obtained itemsets from two algorithms as well as highly decreasing of execution time when applying PMIMA algorithm. Also as appeared in table the number of frequent itemsets increased as minimum support decreased and verse versa and that reflect to the time and number of association rules can be extracted from all or maximal frequent itemsets.

| Minimum Support | Frequent itemsets | Execution time (second) | maximal frequent itemsets No. | Execution time (second) |
|-----------------|-------------------|-------------------------|-------------------------------|-------------------------|
| 40%             | 78923             | 2141.78                 | 24758                         | 413.30                  |
| 50%             | 42277             | 1023.44                 | 10305                         | 76.22                   |
| 60%             | 30804             | 829.84                  | 5357                          | 23.16                   |
| 70%             | 22829             | 581.02                  | 1867                          | 4.22                    |
| 80%             | 18517             | 611.15                  | 443                           | 0.52                    |

### 10.3. Association rules Extraction and classification results

As mentioned previously rules structure consists of rule body represents frequent itemsets in document and rule head represents class of that document as presented in table 2.

| Rule support | Rule body | rule head | Length |
|--------------|-----------|-----------|--------|
| 12%          | ادلب، طلب | Sport     | 2      |
| 12%          | سوق | Economic | 1      |
| 16%          | مصادر، علم، فن، نشر | Law | 3      |
| 14%          | هدف | Sport | 1      |
| 12%          | علم، عمل | Economic | 1      |
| 14%          | نسم، علم، فضاء، فلك | Astronomy | 5      |

Association rules constructing process using maximal frequent itemsets generated by PMIMA algorithm has been evaluated through comparing it with the number of rules extracting by using all frequent itemsets generated by Apriori algorithm. Table (3) shows association rules numbers extracted according to frequent itemsets obtained from two algorithms for sport, astronomy, economic and law categories when minimum confidence equal to 70% and minimum support for generating frequent itemsets equal to various values.
Document classification evaluated according to formula 10 which shows the difference between the accuracy that obtained from rule length, class majority and rule weight classification methods using association rules presented in table 2. Table 3 appears classification results for 320 documents dividing equally among sport, economic, astronomy and law categories i.e. 80 documents for each category with minimum confidence and minimum support of rules equal to 70% and 10% respectively and deferent minimum support for generating all and maximal frequent itemsets.

Table 4. accuracy of documents classification

| Categories | Minimum support of frequent itemsets | Documents accuracy classification of all frequent itemsets generated by Apriori | Documents accuracy classification of maximal frequent itemsets generated by PMIMA |
|------------|-------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
|            |                                     | Rule length | Rule majority | Rule weight | Rule length | Rule majority | Rule weight |
| Sport      | 70%                                 | 0.36        | 0.32          | 0.41        | 0.76        | 0.60          | 0.71        |
| Economic   |                                     | 0.05        | 0.26          | 0.23        | 0.15        | 0.76          | 0.72        |
| Law        |                                     | 0.46        | 0.50          | 0.50        | 0.82        | 1             | 1           |
| Astronomy  |                                     | 0.31        | 0.27          | 0.15        | 1           | 0.98          | 0.98        |
| Sport      | 60%                                 | 0.68        | 0.85          | 0.85        | 0.83        | 0.92          | 0.93        |
| Economic   |                                     | 0.41        | 0.45          | 0.42        | 0.43        | 0.87          | 0.80        |
| Law        |                                     | 0.48        | 0.45          | 0.43        | 0.55        | 1             | 1           |
| Astronomy  |                                     | 0.40        | 0.17          | 0.17        | 0.97        | 0.98          | 0.98        |
| Sport      | 50%                                 | 0.75        | 0.87          | 0.87        | 0.82        | 0.90          | 0.91        |
| Economic   |                                     | 0.06        | 0.66          | 0.66        | 0.22        | 0.93          | 0.92        |
| Law        |                                     | 0.43        | 0.28          | 0.31        | 0.46        | 0.98          | 0.98        |
| Astronomy  |                                     | 0.51        | 0.17          | 0.17        | 1           | 1             | 1           |

Clearly accuracy of classification methods using maximal frequent itemsets according to maximal frequent itemsets obtained from PMIMA overcome the classification results depending on all frequent itemsets generated by Apriori. As shown in table 4, the number of association rules of both algorithms increased as the minimum support decreased for example when set minimum support equal to 50%, 59256 rules have been extracted by Apriori and 3264 rules by PMIMA and this effects the classification results. Regarding outcomes acquired relying on maximal itemsets, rule length classification method appears worst results than other two classification methods.

11. Conclusion

In this research association rule mining extracting from maximal frequent itemsets was used for classifying Arabic documents. In order to get accurate and less volume of text, preprocessing steps is necessary especially with Arabic text because of many challenges related to this language. The number and time required for generating association rules highly depend on predefine minimum support value for producing frequent itemsets. Through using different values of rule minimum confidence and support, can get the best collection of rules to classify documents. Experiments showed classification of Arabic documents according to maximal frequent itemsets better than classification result depending on association rules extracted from all frequent itemsets generated by Apriori using length rule, majority rules and weight rule classification techniques.
References

[1] QASEM A. AL-RADAIDEH, EMAD M. AL-SHAWAKA AND ABDULLAH S. GHAREB, (2011), An Approach for Arabic Text Categorization Using Association Rule Mining, International Journal of Computer Processing Of Languages, Vol. 23, No. 1 (2011) 81–106.

[2] Dumais, S., Platt, J., Sahami, M. & Heckerman, D. (1998), ‘Inductive learning algorithms and representations for text categorization’, ACM, pp.148 – 155

[3] Abdullah Ayedh, Guanzheng TAN, Khaled Alwesabi and Hamdi Rajeh, (2016), ‘The Effect of Preprocessing on Arabic Document Categorization’.

[4] Rehab Duwairi, Mahmoud El-Orfali,(2014), ‘A study of the effects of preprocessing strategies on sentiment analysis for Arabic text’, Journal of Information Science

[5] Rakesh Agrawal, Tomasz Imielinski and Arun N. Swami”, Data Mining: A Performance perspective, IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 6, December 1993, pp. 914–925.

[6] MH Dunham, Y. Xiao, L. Gruenwald, Z. Hossain, A SURVEY OF ASSOCIATION RULES ACM Survey Journal, 2000.

[7] Rakesh Agrawal and Ramakrishnan Srikant, Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the Twentieth International Conference on Very Large Databases, pp. 487-499, Santiago, Chile, 1994.

[8] Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery, 8(1), 53–87.

[9] Hannu Toivonen, Sampling Large Databases for Association Rules, Proceedings of the 22nd International Conference on Very Large Databases, pp. 134-145, Mumbai, India, 1996

[10] Agrawal, R., & Shafer, J. C. (1996). Parallel mining of association rules. IEEE Transactions on Knowledge and Data Engineering, 8(6), 962–969.

[11] Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). Data Mining and Knowledge Discovery, 1(4), 343–373.

[12] Eui-Hong Han, Karypis, G., & Kumar, V. (2000). Scalable parallel data mining for association rules. IEEE Transactions on Knowledge and Data Engineering, 12(3), 337–352.

[13] Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning, pp. 137–142 (1998)

[14] Jingnian Chen, Houkuan Huang, Shengfeng Tian, Youli Qu, (2009), Feature selection for text classification with Naïve Bayes, Expert Systems with Applications 36, 5432–5435.

[15] Vishwanath Bijalwan, Vinay Kumar, Pinki Kumari and Jordan Pascual, (2014), KNN based Machine Learning Approach for Text and Document Mining, International Journal of Database Theory and Application Vol.7, No.1, pp.61-70.

[16] Arlina D’unha, Dr.A.K.Sen,(2015), ‘Hierarchical Approach for Scientific Document Classification’, International Conference on Computing, Communication and Automation.

[17] Lucas Dixon, John Li, Jeffrey Sorensen, (2018), ‘Measuring and Mitigating Unintended Bias in Text Classification’, AIES’18, February 2–3, 2018, New Orleans, LA, USA

[18] AL-RADAIDEH Q. A., AL-SHAWAKA E. M., GHAREB A. S., & ABU-SALEM H. (2011). ‘An Approach for Arabic Text Categorization Using Association Rule Mining’. International Journal of Computer Processing of Languages, 23(01), 81–106.

[19] Cherif W., Madani A., & Kissi M. (2015), ‘A new modeling approach for Arabic opinion mining recognition’, Intelligent Systems and Computer Vision (ISCV).

[20] Kanan, T., & Fox, E. A. (2015), ‘Automated arabic text classification with P-Stemmer, machine learning, and a tailored news article taxonomy, Journal of the Association for Information Science and Technology, 67(11), 2667–2683.

[21] Abdullah Ayedh, Guanzheng TAN, Khaled Alwesabi and Hamdi Rajeh, (2016), ‘The Effect of Preprocessing on Arabic Document Categorization’.
[22] Al-Anzi F. S., & AbuZeina D. (2017), ‘Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing’, Journal of King Saud University - Computer and Information Sciences, 29(2), 189–195.

[23] Said B., Abdellh H., & Mohamed K., (2017), 'Arabic text classification using new stemmer for feature selection and decision trees'. Journal of Engineering Science and Technology Vol. 12, No. 6.

[24] Al-Radaideh Q. A. & Al-Abrat M. A. (2018). ‘An Arabic text categorization approach using term weighting and multiple reducts’. Soft Computing.

[25] Bahassine S., Madani A., Al-Sarem M., & Kissi M. (2018). 'Feature selection using an improved Chi-square for Arabic text classification'. Journal of King Saud University - Computer and Information Sciences.

[26] Taouil, N. Pasquier, Y. Bastide, and L.Lakhal, 'Mining bases for association rules using closed sets'. In IEEE Intl. Conf. on Data Engineering, March 2000.

[27] Ye-In Chang, Chia-En Li, Wei-Hau Peng and Syuan-Yun Wang, 'Efficient superset-lattice algorithms for mining closed frequent itemsets and maximal frequent itemsets in data streams', international journal of electrical and engineering, vol.20, no.2 pp.51-63(2013).

[28] http://sourceforge.net/projects/ar-textmining