A Different Text Mining Process for Classifying Journal Databases Using Machine Learning Algorithms

G. Thailambal, Ananthi Sheshasaayee

Abstract--- ***Google is the information repository for the entire world and is an important Search engine used for Information Retrieval. Accessing web pages is getting increased everyday which can be compared to the speed in which light travels. Biggest Challenge is identifying the user interest and providing them information based on the high relevancy. Mostly researchers search journal documents for their research every day. Classifying the content as papers or Slides or thesis is very difficult as the words used in these documents are not semantically checked. To mine the correct content in web page Data Mining is used by most of the researchers. Text Mining is one of its application. Text mining in nutshell is extracting useful information from unstructured data. The proposed Model Author Keyword Weightage in Journal Ranking (AKWJR) is developed to retrieve relevant journals that will help the researchers to identify the relevant documents from the pool of irrelevant documents. In many keyword ranking applications such as RAKE and TEXTRANK author annotated keywords were compared and used for ranking. The assignment of keywords to article by the author is different in their form and perspective. Though they were not choosing the keywords in a controlled vocabulary the keywords were used to describe their own content in the article. Two algorithms were used to arrange the keywords according to topics and the keywords inside the journals will be scored depending on its presence in various fields in the article. Depending on the score the journals will be ranked in such a way that the author can decide whether to open the article for their requirement. This is achieved through Latent Dirichlet Allocation, RankSVM and TF-IDF Algorithms.***

I. INTRODUCTION

Web mining access useful information from sources such as web structure, hyperlinks, page content, usage data and logs. It is classified into three major categories, namely, *Web Structure Mining*, *Web Content Mining* and *Web Usage Mining*.

- Web Structure Mining ascertains information from Hyperlinks. It is useful web page for the search query and information about communities that share common interests.
- Web Content Mining extracts information from web page contents. It classifies and cluster web pages, Customer reviews, forums and discover patterns through the web pages.
- Web Usage Mining detects user access patterns from usage logs [12].

Web Content Mining

Web content data consists of unstructured text data and it requires both data mining and text mining techniques. Applying data mining techniques to unstructured text is Knowledge Discovery in text (KDT) or text data mining or text mining [13].

Text Mining

Text mining extracts useful information from unstructured data as they summarize the words in the document. Word is the main unit of text, which provides meaning to the information. The words are combined to form a sentence through which the information is specified clearly. In text mining, the documents are used as prime source of analysis of information. Most of the text are in unstructured form due to the absence of syntactical structures. The prime issues faced while handling the words are Polysemy and Synonymy. The polysemy gives multiple meanings for a word. Synonyms gives same meaning with different words. Thus text mining is an effective way for documentation. Information Retrieval (IR) is a process that retrieves useful data from large sources of data. Web browser is a software that supports search engines to view and retrieve the data. It plays a vital role in text mining.

Preprocessing

The preprocessing in the Text mining consists of *Tokenization*, *Stop word removal*, *Stemming*, and *Word normalization*. Tokenization divides text into small tokens by removing white space, commas, semicolon, quote and period. Stop word removal is separation of text based on grammatical values such as noun, verb, pronoun, article, conjunction, preposition, numbers and alphanumeric. Normalization converts the words with same meaning into one form. Stemming removes all prefixes, infixes and suffixes to reduce the word to its roots. For example, Teacher, Teaches, Learner derived from the stem 'Teacher' which needs to be considered as 'Teach' for reducing the dimension of the word [14]. According to Cristian Moral, stemming obtains the root of a word by clearing the affixes that carry grammatical or lexical information about the word [15].

II. LITERATURE REVIEW

Chengzhi Zhang et. al, [1] proposed a model called *Conditional Random Field* (CRF) for automatic keyword extraction to identify the keywords in a document.

Manuscript received September 16, 2019.

Dr.G. Thailambal, Associate Professor, Department of Computer Science, Vels Institute of Science, Technology and Advanced Studies, Chennai, T.N, India.. (e-mail: thaila.research@gmail.com)

Dr. Ananthi Sheshasaayee, Associate Professor &Head, PG& Research Department of Computer Science, Quaid-E-Millath Government College for Women (Autonomous),Chennai, T.N, India. (e-mail: ananthi.research@gmail.com)

Documents were collected from database of Information center for social science of RUC. The authors randomly chose 600 academic documents in the Economics field and a 10-fold Cross-validation was done. Pre-processing such as POS tagging and sentence segment was done in all the documents. Feature vector was extracted and keyword was annotated with one kind of labels such as ‘KW-B’, ‘KW-I’, ‘KW-S’, ‘KW-N’, ‘KW-Y’, which was used for training the CRF model. The researchers were able to predict new document using this model. Contingency table, Precision, Recall and F1-Measure was used for evaluation. They observed that CRF model outperformed other approaches such as SVM and Multiple Linear Regression models.

Marina Litvak et. al, [2] proposed and compared supervised and unsupervised approaches for identifying keywords that can be used in extractive summarization of text documents. In supervised approach, the authors employed classification algorithm and in unsupervised approach, HITS algorithm was used. Graph based feature was used in which In-Degree, Out-Degree, Frequency, Frequent word distributions, location Score, TF-IDF Score, Headline score was calculated. Further, the feature selection ratio was calculated based on Gain Ration value. The authors used the news articles provided by document conference 2002 in which 566 English texts in a collection taken for research as datasets. Weka software was used with J48, Support Vector Machine, and Naïve Bayes in built algorithms applied for classification of *Yes* or *No*. AUC were visualized. For unsupervised model, HITS algorithm was executed several times and was compared with visualized using AUC (Area under curve). The authors concluded that if large number of documents were used, supervised classification had better performance.

Ian H. Witten et. al, [3] proposed KEA an algorithm that automatically extracts key phrases from text using lexical methods. The researchers calculated feature values for each candidate and a machine learning algorithm was used to predict the candidates with good phrases. Datasets were used from computer science technical reports, which contained 46000 documents. The authors performed four experiments, namely, *KEA’soverall effectiveness, the effect of changing the size and source of global corpus, the effect of changing the number of training documents, KEA’s performance using abstracts rather than full text.* For each training document, candidate phrases were identified and their feature values were calculated. KEA uses Naïve Bayes techniques, which uses two set of weights Positive (Key phrase found) and Negative (Key phrase not found). Two special features were used such as first occurrence of phrase and discretization table. The author concluded that KEA outperformed in summarizing, browsing, searching and clustering where manual key phrase was infeasible.

Sabah Mohammed et.al. [5] Spam Filtering method eliminates unwanted mails which has many different types of filters such as Word Lists filter in which simple and complex list of words given in spam, Black lists and white lists filter contain known IP addresses of spam and Non-spam senders, Hash Tables filter summarize emails into pseudo-code values and repeatedly sightings of hash values in bulk mailing. New types of spam filtering relies on statistical features of spam which scan and analyse complete mail lists to identify whether it is spam or not. Every new mail is compared with database of spam mails and finds whether it is spam or not. The author accepts that this method generally pushes spam detection ratio to higher percent and even against Phishing attempts.

Rajini Jindal et. al., [6] framed research questions to select papers. They were assessed for relevance and were either included or excluded in the research. A total of 132 relevant studies for text classification was found. A comparative analysis was made for different feature selection methods used by different authors, different document representation methods and different data mining methods. Dataset used was text classification papers from 1999 to 2012, published in conference proceedings and journals of high reputation. Finally, the authors concluded that most of the researches used UCI (University of California Irvine) repository and employed the vector space model. Machine learning models have better features than statistical models and SVM, KNN algorithms were widely used Machine learning algorithms. The authors also have presented the most important text classification journal, years showing maximum publications, distribution of papers after years 2004, important data mining methods used, most important feature selection methods used, widely used datasets and distribution of document representation methods.

Levis Teixeireset. al., [7] illustrated how the extraction of topics was made with the dataset collection of one lakh texts from Portuguese, English and Czech Languages. Multi words were extracted from Local max algorithm and suffix. Arrays were used for word extraction and prefixes counting. 25 best ranked terms for each one of the six measures in which assigned a classification (Good topic descriptor) G, (Near good topic descriptor) NG, (Bad topic descriptor) B, (Unknown) U, and (Not Evaluated) E. K-statistics was used to measure the degree of agreement between evaluators. Least operator, Least Median operator and Least Bubbled Median Operator were used. The authors concluded that Bubbled variant showed interesting results for three long users especially for Portuguese and Czech. Least ad least Median Operator were best for English.

Marine Sokolovaet. al., [8] analyzed 24 performance measures used in Machine Learning Classification tasks (i.e.) Binary, Multiclass, Multi-Labelled and Hierarchical. The evaluation of classification results depended on the invariance properties of the measure. The effects of change in the Confusion matrix were studied. A reliable evaluation was performed to employ measures such as representativeness of class distribution, reliability of class labels, unimodal and Multi-modality of classes. The authors stated that classification of human communications differ from document classification and they require different performance measures.

Feifan Liu et. al., [9] have explored different keyword extraction algorithms using transcripts of ICSI meeting corpus.

A graph based algorithm was developed to leverage global information and reinforcement from summary sentences. Various performance measures using individual F-Measures and a weighted score relative to the system performance was performed. Weighted Graph model such as word to word connections, sentence to word connection and sentence to sentence connection was used. The authors chose top 5 words as keywords for a topic. Human annotated keywords were used for reference and used F-measure and Pyramid for evaluation. They observed that TF-IDF method was highly competitive. Further, the authors found that the human evaluation results were consistent with the automatic evaluation metrics in terms of ranking of different systems.

Martin Dostalet. al., [10] proposed an experimental approach to automatic key phrase extraction based on statistical methods and Wordnet-based pattern evaluation. Key phrase candidates were extracted, derived from combination of graph methods. Text Rank and Statistical TF-IDF method. Keyword candidates merged with NLPOS (Part-of-Speech) pattern text. Text preprocessing, keyword and key phrase extraction was established by the author to remove non-significant character. POS-patterns for 3-Grams and 2-Grams with tags such as Noun, Verb, and Adjective were used for key phrase extraction. A collection of newspaper articles from the web was used as the dataset. Further, human annotators were used for manual tests and the author achieved 37.4 % precision and 54.6% recall for small corpus which gets reduced for higher number of corpus documents.

PuWanget. al., [11] overcame the shortages of BOW approach by embedding background knowledge from Wikipedia into a Semantic Kernel. The authors used four real datasets, namely, Reuters 21578, OHSUMED, 20 Newsgroups and Movies to evaluate the performance. Preprocessing of the documents were made by eliminating stop words, pruning words and stemming. Thesaurus was used for enriching wiki documents with Hyponyms, synonyms and associate concepts. Semantic Kernels were also added and Precision was calculated. Wiki-SK provides higher micro- and macro-precision values on all datasets. For identifying eligible candidates, cosine similarity between TF-IDF kernels based methods were used to model relevant multiword concepts as individual features and to assign meaningful strength through proximity matrix.

III. PROPOSED METHOD & RESULTS

The model AKWJR structure, applies the machine learning algorithms, namely, Latent Dirichlet Allocation (LDA) for topic extraction with semantically analyzed words, calculate TFIDF and keywords weightage in the documents. RANKSVM algorithm is used to learn preferences in this research.The following is the process involved in collection of documents and pre-processing:

- Dataset: In this research, a total of 10000 Journal Documents in PDF format has been extracted and stored from many journals in computer science discipline. 8000 documents were considered for training and the remaining documents for testing. The documents were predominantly retrieved from journals such as Science Direct, IEEE and DOAJ.
- Pre-processing: For the pre-processing of the documents, the Python SciKit Learn tool has been used. During this stage, Non-journal documents and documents without keywords and other document types apart from PDF format were removed from the dataset. Preprocessing is an important task, as this process is used to remove unwanted information from the document in order to reduce the document size and appropriate content will be considered for the processing. 30 Stemming is the process of reducing to its root word, which helps to reduce the size of the document. However, it is not necessary that the stem need to be identical to the morphological root of the word. For example, a word in the document ‘Learning’, ‘Learned’, ’learn’ can be treated as ‘Learn’ which does not affect the meaning of the document. Each document is applied with this stemming process. In order to support phrase search, stop words are removed as they will not give meaning to the word. From each document the words like ‘a’, ‘an, ‘the’ will be removed in order to decrease the size and save the execution time.

Apply LDA Topic Modeling

In the implementation of ***Latent Dirichlet Allocation***, LDA automatically identifies the topics from documents. The primary purpose is the inference of words in the sentences along with its word count and the proportions of word usage in individual topics can be obtained. The algorithm uses the following three steps:

Step 1: Identify the number of topics to be divided.
Step 2: Assign every word to a topic.
Step 3: Identify how relevant a word to a topic assigned and how relevant the topic distribution in the document.

Each document can be viewed as different topics using LDA model, which follows supervised machine learning technique. LDA is represented as Plate Model in which repeated variables are represented as Rectangles and Circles. Figure 1illustrates Plate notation of LDA.

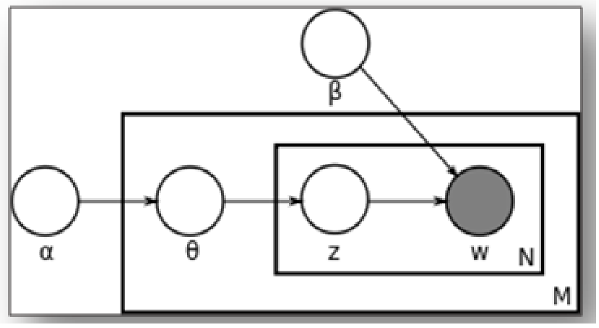


Fig.1: Plate Notation of Latent Dirichlet Allocation

Where α ***i***s the parameter of the Dirichlet prior on the per-document topic distributions,
βis the parameter of the Dirichlet prior on the per-topic word distribution,
$\theta_m$is the topic distribution for document *m*,
Z_{mn} is the topic for the *n*-th word in document *m*, and

$\mathbf{W}_{mn}$is the specific word,
W is the only variable that can be observed and directly measured while other variables cannot be observed easily.

Tag the Documents

The TF-IDF (Term Frequency – Inverse Document Frequency) denotes the importance of a word to its document in a corpus and is widely used in the Information Retrieval.

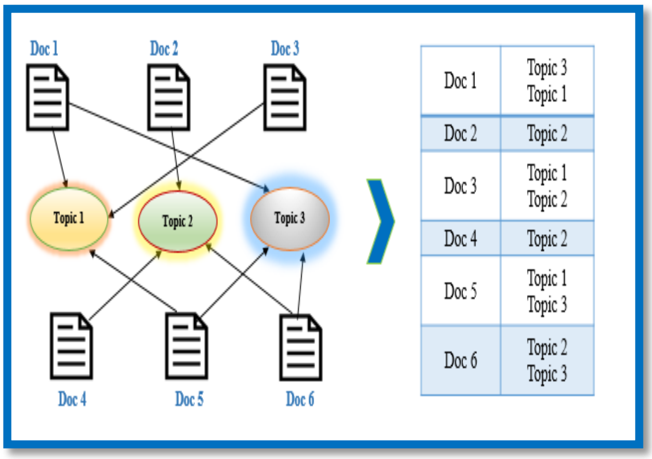
The TF-IDF weightage of words determines, whether a word can be tagged to the document or not. Tagging is the process of relating the topic to the document. The relation is based on finding top 5 words in the TF-IDF, whose value should be greater than the threshold (i.e.) 50% of the TF-IDF value.

Term Frequency refers to the number of times a word appears in a document and Inverse Document Frequency is the logarithm of the number of documents in a dataset divided by number of documents under specific term. Every document is of different length and the term may appear many times in a long document compared to the shorter documents. Similarly, some terms are not frequent such as 'is', 'was', 'at' and hence, those terms need not be considered for calculation. Figure.2 illustrates how the documents 1, 2, 3, 4, 5 and 6 tagged to Topics 1, 2 and 3.

$$\text{TF}(t) = \text{F}(t, d) / (\text{Number of words in } d) \quad (3.2)$$

$$\text{IDF}(t, D) = \log N \mid \{d \in D : t \in d\} \mid \quad (3.3)$$

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d).\text{IDF}(t, D) \quad (3.4)$$



Doc 1	Topic 3 Topic 1
Doc 2	Topic 2
Doc 3	Topic 1 Topic 2
Doc 4	Topic 2
Doc 5	Topic 1 Topic 3
Doc 6	Topic 2 Topic 3

Fig.2: Tagging Documents to Different Topics

Extract the Keywords and Rank the Documents Using Ranksvm Algorithm

RANKSVM algorithm helps to identify the relevancy of results obtained for a query.

Some features are mapped with the query results and they are used as training data. Mapping the similarities between query and feature space, finds the distance between them for optimizing the problem. Suppose C is a dataset containing elements C_i r where r is the ranking method applied to C. If the ran-k of C_i is higher than rank of C_j then its position is value 1 otherwise value 0. This proposed work uses pairwise ranking since it minimizes ranking loss when compared with other point-wise and list-wise approaches. Figure illustrates the ranking model.

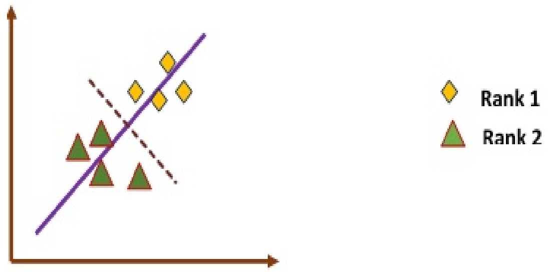


Fig.3: Ranking Model

Figure.3 illustrates Rank 1 higher than Rank 2. A dotted line distinctly indicates the ranking of the documents. Linear SVM training is similar to RankSVM training. According to O. Chapelle [4], the training samples x*i* simply need to be replaced by the differences x*i* − x*j*for (*i, j*) ∈ *P*. In matrix form, this means replacing the matrix *X* by *AX* where *A* is a *p* × *n* sparse matrix, *p* = |*P*|. Each row of *A* encodes a preference: if (*i, j*) ∈ *P*, there exists a row *k* of *A* such that *Aki* = 1, *Akj*= −1 and the rest of the row is 0. [4]. Ranking is divided into three types Pair-wise, List-wise and Point-wise. Pairwise accuracy is mainly concentrated since it is directly related to the loss term of RankSVM and is used in Statistical and Medical data analysis with the name concordance index or Kendall's *τ*.

Performance Evaluation Of Training And Testing Data

Figure.4 and Table 1 illustrate the performance evaluation of training and testing data separately. 80% of the training data is taken and 20% of testing data is taken for calculation and it shows Precision, Recall, F-Score and Accuracy of both the data.

Table 1: Performance Evaluation of Training and Testing Data

Number of Docs	Execution Time
1k	20.76
5k	103.8
8k	166.08
10k	207.6

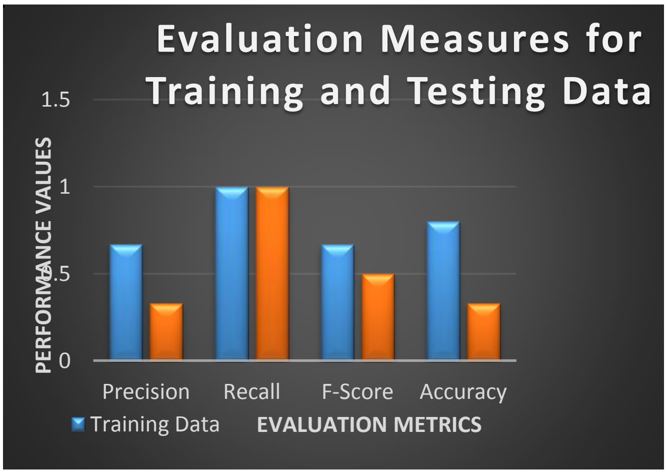


Fig.4: Performance Evaluation of Training and Testing Data

Execution Time

Table.2 explains how the execution time differs for increasing number of documents and the Figure.5 illustrates that when the number of document is higher the execution time also increases, since the whole document needs to be taken for score evaluation. But comparatively when other methods such as RAKE and KEA are analyzed they use only abstract for the score calculations.

Table 2: Execution Time of Different Document Length

Data	Precision	Recall	F-Score	Accuracy
Training Data	0.67	1	0.67	0.8
Testing Data	0.33	1	0.5	0.33

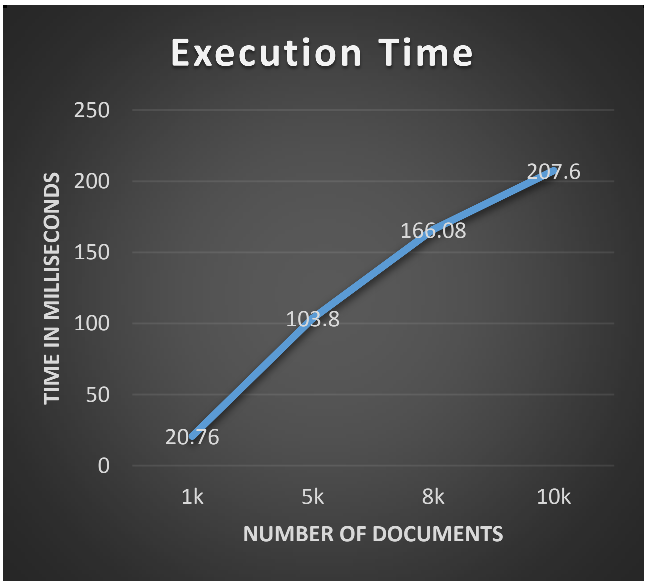


Fig.5: Execution time of Documents

IV. CONCLUSION

The model **AKWJR** spurs new ideas for researchers and relevance in search. The research documents is better extracted when visualized from user's perspective. It improves the research quality by extracting useful information from journals. Mining from large volume of text is refined by giving appropriate value to the text documents. This research work has applied citation value instead of journal selection for better results because the citations vary based on the accessibility of journals. The documents are added correctly, which avoids tagging wrong documents for an appropriate query provided by the researcher. The model **AKWJR** reduces the time consumption and produces appropriate query results. In the searching process, the researchers have limited access to the entire document, therefore, this research work facilitates the users with relevant document that benefits their nature of work. In this model **AKWJR,** any number of journals can be classified based on the semantic analysis of words in documents and are ranked based on the keywords used.

REFERENCES

1. Zhang, Chengzhi& Wang, H & Liu, Y & Wu, Dan & Liao, Y & Wang, B, "Automatic keyword extraction from documents using conditional random fields". Vol. 4. No. 3, pp.1169-1180, June 2008.
2. Marina Litvak, Mark Last, "Graph-Based Keyword Extraction for Single-Document Summarization", Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization, pages 17–24, Manchester, Aug. 2008.
3. Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, Craig G. Nevill-Manning, KEA: practical automatic keyphrase extraction", Proceedings of the fourth ACM conference on Digital libraries, pp.254-255, Berkeley, California, USA, 11-14, Aug 1999.
4. O. Chapelle, S.S. Keerthi, "Efficient Algorithms for Ranking with SVMs", Information Retrieval, Vol. 13, No. 3, pp. 201, 2010.
5. Sabah Mohammed, Osama Mohammed, Jinan Fiaidhi, Simon Fong, & Tai Hoon Kim, "Classifying Unsolicited Bulk Email (UBE) using Python Machine Learning Techniques" International Journal of Hybrid Information Technology, Vol. 6, No.1, pp. 43-56, Jan. 2013.
6. Rajini Jindal, Ruchika Malhotra, Abha Jain, "Techniques for Text Classification: Literature Review and Current Trends", Webology, Vol. 12, No. 2, Dec. 2015.
7. Teixeira L., Lopes G., Ribeiro R.A. (2011) "Automatic Extraction of Document Topics", In: Camarinha-Matos L.M. (eds) Technological Innovation for Sustainability. DoCEIS 2011.IFIP Advances in Information and Communication Technology, Vol 349. Springer, Berlin, Heidelberg.
8. Marina Sokolova, Guy Laplame, "A Systematic Analysis of Performance Measure for Classification Tasks", Information Processing and Management, Vol. 45, No. 4, Elsevier, pp:427-437.
9. Feifan Liu, Deana Penell, Fei Liu and Yang Liu, "Unsupervised Approaches for Automatic Keyword Extraction using Meeting Transcripts", Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pp. 62-628, Boulder, Colorado, June 2009.
10. Martin Dostal and Karel Jeˇzek, "Automatic Keyphrase extraction based on NLP and Statistical Methods", Proceedings of the Dateso 2011: Annual International Workshop on Databases, Texts, Specifications and Objects, Pisek, Czech Republic, PP. 140-145, ISBN 978-80-248-2391-1, April 20, 2011.
11. Pu Wang and CaroletteDomeniconi, "Building Semantic Kernels for Text Classification using Wikipedia", Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 713-721, Las Vegas, Nevada, USA, Aug. 24-27, 2008.
12. Bing Liu, "Web Data Mining Exploring Hyperlinks, Contents, and Usage Data", Second Edition, Springer, 2011.
13. Anurag kumar, Ravi Kumar Singh, "A Study on Web Content Mining", International Journalof Engineering and Computer Science, ISSN: 2319-7242, Vol. 6, Issue 1, pp.20003-20006, 2017.
14. Ayedh, A.; TAN, G.; Alwesabi, K.; Rajeh, H. "The Effect of Preprocessing on Arabic Document Categorization". *Algorithms*, *9*, 27, 2016.
15. Cristian Moral, Angélica de Antonio, Ricardo Imbertand Jaime Ramírez. "A Survey of Stemming Algorithms in Information Retrieval", Information Research, Vol.19, No.1, March, 2014.