Expanding Queries in the Information Retrieval System Using Stemming Approach

Yellepeddi Vijayalakshmi, B. ArunKumar, & G. K. D. PrasannaVenkatesan

Abstract Numerous approaches have been made to find solutions for constructing text information stemmers. These stemmers are generally utilized in perception of application-oriented projects, specifically when they deal with the development of information retrieval (IR) schemes. Moreover, text stemming, as an approach for stripping sets of their suffixes or prefixes is considered as task suffering, when there are problems like single solution, vocalization ambiguity, incorrect removal and so on. However, many investigators claim that stemming approach has reached a high level for accuracy and precision while retrieving texts. In some cases, these stemmer algorithms are measured as black boxes, and it is not probable to access either source code or for corpora to validate estimation, which is used to attain accuracy. As stemmer algorithms are extremely significant for researchers, its comparison and estimation are more significant to facilitate choice of stemming approach to use in information retrieval. Here, stemming algorithm is anticipated based on feature reduction from multi-source-Ant Colony (MS) by performing word bunching and scoring (W-CS), so as to provide solutions to the drawbacks mentioned above. Here, an automatic approach for pre-processing the text has been carried out for evaluation of and comparison of text retrieval from queries that consider performance metrics like accuracy and evaluation period for the stemming algorithm. Simulation was carried out in MATLAB environment. The suggested model outperforms prevailing approaches in terms of accuracy.

Keywords Information Retrieval, text retrieval, Multi-source, bunching, scoring, stemming

I INTRODUCTION

In Information Retrieval, numerous challenges have been considered for information requirement such as Query and automatic scoring of individual documents which are based on relevance information [1]. Diplomatic instances may include Probabilistic indexing approach that analyzes queries that are allocated to documents. This function deliberates scoring with the log-ratio of relevance probability [2]. However, based on various facts and practical circumstances relevance information is not available, where improvements are measured with text statistics validation. Documents are then constructed with the link via those statistics [3]. For instance, scoring approaches like Vector space modelling, TF, IDF and Divergence from Randomness (DFR) approach have been constructed [4]. A sensible approximation of RSJ model leads to development of famous BM25 scoring functionality [5]. Significant approximation in probabilistic modelling is to develop "language model' of document and evaluate its likelihood for providing query execution; query language model comes under Kullback-Leibler divergence-sourced loss functionality [6].

Indeed, of various efforts of information retrieval (IR), in case of validation phase, numerous IR tasks provide validation criteria that move beyond essential counting of number appropriate to documents in ranked list [7]. IR measurement using diverse metrics is more crucial due to its retrieval objectives. Investigators have to attain various perceptions of retrieval significance [8]. While these preferences move strongly towards prior retrieval documents, Mean Reciprocal Rank (MRR) is measured as superior metrics, while if investigators attempt to attain broader retrieval performance summary determination of Mean Average Precision (MAP) is more appropriate [9]. Therefore, there is a gap amongst decision processing of retrieval model, and ultimate validation criterion is anticipated to determine achievement in task [10]. Specifically, it is enviable to deal with retrieval systems used for certain IR efficiency metrics. However, various IR investigators have previously initiated this opportunity. One amongst the effectual cases is stemming. It openly handles stemmers for training data by bypassing steps of evaluating relevance states of various documents. Based on this paradigm, various attempts have been made to openly optimize IR metrics like MAP and NDCG (Normalized Discounted Cumulated Gain). Moreover, it is considered that certain evaluation metrics are less informative than others. As stated in [10], certain IR metrics therefore do not essentially summarize training data as well, if optimizing IR metrics are initiated from training data. Statistics attained from data may not be completely cast off and explored as in fig. 1.Certain essential opposition direction is to concentrate on designing scoring functionality of documents, however with acknowledgement of diverse retrieval objectives and finally ranking context. In [10], a model termed as 'less is more' is anticipated as one of the instances. This model considers retrieval documents previously attained as irrelevant while computing document relevance for computing present ranking position; this algorithm is considered as equivalent model to enhance Reciprocal Rank Measure. Various generalized and flexible determination with respect to this model is anticipated. In this framework, Bayesian decision theory is used to integrate diverse ranking schemes via pre-defined loss functionality. IR models' outcome, however is deficient in providing ability to directly handle IR metrics into rank decision. Here, this paper concentrates on stemming with Multi-source Ant Colony (MS-AC) so as to reduce text to its root or stem by considering prefixes or suffixes and eliminate them based on Text Bunching. The outcomes are determined as token. While processing the outcomes, token moved by stemming algorithm provides no meaning; thereby scoring functionality is used to transform tokens to texts in dictionary. In various application field, stemming model is extensively utilized for IR, where scoring is significantly related to it for machine translation purposes and needs appropriate words and not tokens.

- ***Mrs. Yellepeddi Vijayalakshmi*** *is currently pursuing Doctoral degree program in Computer Science & engineering, Karpagam Academy of Higher Education (Deemed to be University), India, PH-8281768366. E-mail: vijayasasi11@gmail.com*
- ***Dr.B.ArunKumar*** *is currently serving as Associate Professor in Computer Science & engineering, Karpagam Academy of Higher Education (Deemed to be University), India.*
- ***Dr. G. K. D. PrasannaVenkatesan*** *is currently serving as Dean, Faculty of Engineering, Karpagam Academy of Higher Education (Deemed to be University), India.*

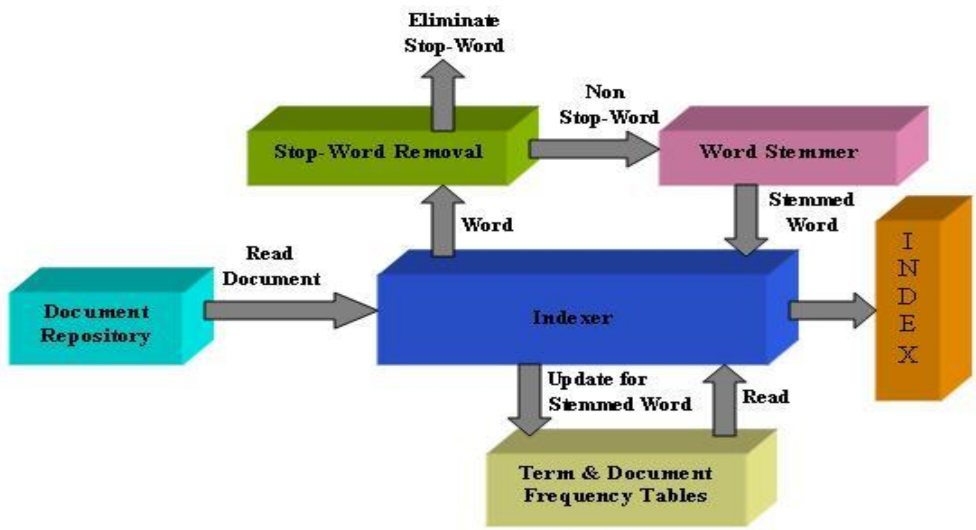

Fig 1: Information retrieval flow diagram

The paper is structured thus: Section II explains background modelling, Section III depicts anticipated model based on Multi-Source Ant Colony based on bunching and scoring for text retrieval, Section IV provides numerical results and discussions associated with anticipated model and Section V concludes with advice of future direction for further investigations.

II RELATED WORKS

Shyam et al. [11] have depicted experimental outcomes that recommend hybrid modelling with light and statistical stemmers in more appropriate stemming algorithm for Arabic language. Meseh et al. [12] have talked of CHI square model as feature selection approach. Duwari et al. [13] have offered and evaluated three feature reduction approaches that are used in Arabic texts. These approaches include stemming model, clustering and light stemming. Harag et al. [14] have cast off Singular Value Decomposition (SVD) to choose more appropriate features for classification purpose. Zahran et al. [15] have executed Radial Basis Function (RBF) networks sourced on Particle Swarm Optimization (PSO) approaches as feature selection model to compare document frequency and Chi-square statistics algorithms. In background models, ranking algorithm is measured for ordering of output results from most of least used items [16]. In general, ranking is sourced on frequency and location; documents with superior term occurrences are ranked highly. A sensible instance is PageRank algorithm [17], which specifies page relevance with link analysis. Relevance feedback modelling is sourced on the idea that novel query follows the modified version of a prevailing one, attained by rising weight of terms with appropriate items, and diminishing weight of terms in non-appropriate items [18]. So, to overcome various limitations of conventional keyword-based search engines, fuzzy based rule generation is also provided [19]. Here, typos, synonyms are validated with similarity terms with current indexed tokens to offer more appropriate outcomes [20]. Here, stemming with optimization approach is cast off for enhancing performance metrics.

III PROBLEM FORMULATION

In this paper, Multi-source IR is considered for formulating problem, which is converted into optimization problem and the collected text information is chosen in a manner to maximize overall coverage of content with Ant colony optimization. This optimization crisis is modelled as below: Consider 'T' be set of input text document which has to be retrieved and each document has to be split into sentences based on queries. Thereby, 'T' can be re-written as $T = \{S_1, S_2, \dots, S_{|T|}\}$, where $|T|$ is considered as total amount of sentences in 'T' and S_i specifies sentence $i(1 \le k \le |T|)$. This massive MS-IR problem inflicts production of series of sentences; Summary 's' with maximum text length is 'L' by choosing number of sentences from 'T' where overall information coverage of 'S' is determined to be maximized. Generally, optimization is specified as in Eq. (1):

$$S = \max(\sum_{s_i \in D} (CC_s . L_s)) \quad (1)$$

Such that $\sum_{s_i \in D} (CC_s . L_s) \le L$ (2)

Here, CC_s and L_s specifies score-based content coverage and a series of length of sentences 's' respectively. Binary variable L_s is determined to be 1; if s_i is sentence summarization content and zero else. Content score of every sentence is sourced on contained text weight. Moreover, to enhance information saliency and coverage along with reduced redundancies, content text weight should not be considered by other sentences that are chosen already as part of 'S', its weight W_T is summed up to total content coverage score. Henceforth, entire content coverage score 'S' can be computed by accumulating weight of text. Weight is formulated as below in Eq. (3):

$$\sum_{s_i \in D} (CC_s . L_s) = \sum (b_T . W_T) \quad (3)$$

Binary variable b_T is formulated as in Eq. (4):

$$b_T = \begin{cases} 1 & if \sum_{s_i \in D} (d_{kT} . L_s) \ge 1 \\ 0 & otherwise \end{cases} \quad (4)$$

Where d_{kT} is constant that is lesser than or equal to 1; if sentence 'k' contains text 'T' and 0 else.

IV PROPOSED METHOD

In this section, the anticipated MS-AC approach is explained in detail along with text clustering and scoring approach to model stemming algorithm. Experimentation is carried out in MATLAB with windows XP, Intel core 2 processor, with 3 GB memory, 2.66 GHz.

a. Pre-processing

For text-based information retrieval, pre-processing has to be done considering dataset for evaluation. Here, various pre-processing steps are used to text before performing text grouping process. Initially, Stanford core NLP [16] is used for text-based sentence tokenization and segmentation process. In segmentation phase, text is broken down to sentences while tokenization specifies text of each sentence. Secondly, stop word removal [17] is cast off to eliminate concurrently-occurring text that possesses low semantic weight. Stop word list from SMART information retrieval system II [18] is used for English text, and some common stop-word lists are offered in [19] for other text IR. At last, stemming step is utilized to attain the stem of every word, with Ant colony optimization for the corresponding English text.

b. Text scoring

In this phase, score of every text is evaluated by utilizing the approach anticipated in [20]. This model casts off iterative reinforcement model integrating concepts of bi-graph scoring algorithms: PageRank [21] and HITS [22]. It initiates by constructing Tri-graph. Firstly, bipartite graph associates every word with sentences where text appears, where its edges are provided in score-based cosine similarity measure and TF-ISF [23] scores. Secondly, the next graph specifies relationship between every pair of sentences associated with cosine similarity measure and TF-ISF scores. Thirdly, last graph shows relationship between pairs of words with long common sub-strings.

These text scores are evaluated with the utilization of Eq. (5) and Eq. (6) given below:

$$x^{(n)} = \propto \widetilde{U}^T x^{(n-1)} + \beta \widehat{W}^T y^{(n-1)} \quad (5)$$

$$y^{(n)} = \propto \widetilde{V}^T x^{(n-1)} + \beta \widehat{W}^T x^{(n-1)} \quad (6)$$

Where x and y are two matrices that captures text scores and also sentence scores correspondingly. $\widetilde{U}^T$, $\widetilde{V}^T$ and $\widehat{W}^T$ are determined as normalization factors of U, V and W correspondingly. $\widehat{W}^T$ is transposed normalization version of $W.x(n)$ and $W.y(n)$ are determined as matrix 'x' and matrix 'y' values at 'n' iterations. At last, $y^{(n-1)}$ and $x^{(n-1)}$ are determined as matrix 'x' and matrix 'y' at iteration $n-1$. Here, text score is evaluated and available for generating text summary.

Algorithm 1: Text pre-processing

Input: number of text document to maximize sentence summary length
Output: text summary initiation

Step 1: Initiate pre-processing
Step 2: Segmenting text to sentence
Step 3: Perform sentence tokenization
Step 4: Eliminate stop words
Step 5: Substitute every text with its stem text scoring
Step 6: Construct Tri-graph, i.e. text-to-text, sentence-to-sentence and sentence-to-work graphs
Step 7: Use stemming approach
Step 8: Extract sentence summary
Step 9: Construct graph with input text for text grouping
Step 10: Perform MS-AC for IR based of text query

c. Text grouping

Here, two diverse performance metrics are determined for text grouping; they are: purity and entropy. Assume, $'C'$ is original text classes, $'g'$ is grouping of text grouping. Entropy $'E'$ of grouping $'g'$ with size $'S'$ is evaluated as in Eq. (7):

$$E(g_s) = \frac{1}{\log c} \sum_{i=1}^{c} \frac{n_s^i}{n_s} \log \frac{n_s^i}{n_s} \quad (7)$$

Where n_s^i is total amount of document included for text IR in i^{th} class that are allocated to $'g'^{th}$ group. Subsequently, entropy of entire grouping is evaluated as in Eq. (8):

$$E = \sum_{s=1}^{k} \frac{n_s}{n} E(g_s) \quad (8)$$

In general, if entropy value is lesser, then grouping of text will be easier. Also, text group purity is depicted as in Eq. (9):

$$P(g_s) = \frac{1}{n_s} \max_i (n_s^i) \quad (9)$$

Then, purity of complete group is depicted as in Eq. (10):

$$P = \sum_{r=1}^{k} \frac{n_s}{n} P(g_s) \quad (10)$$

Here, if purity value is higher, then text grouping will be effectual.

Algorithm 2: Text Grouping
Input: Tri-graph representation with input text of maximum length
Output: text retrieved
Step 1: Begin
Step 2: Initialize parameters
Step 3: While maximum accuracy is not reached do
Step 4: Identify text to be retrieved from sentences
Step 5: Activate graph labelling
Step 6: Repeat
Step 7: For every sentence in input text document
Step 8: Choose next input text document for computation
Step 9: If graph cannot include more input text document
Step 10: De-activate graph computation
Step 11: Increase total amount of iterative graph
Step 12: Update text scoring to measure unvisited document text
Step 13: Update text length and text score based on binary weight

Step 14: End
Step 15: End
Step 16: For all active computation do
Step 17: Use metrics for updation
Step 18: End
Step 19: Till de-activation stage
Step 20: Increase number of iterations
Step 21: End
Step 22: Update rule for attaining best text IR from generated query
Step 23: If entropy value is lesser than 1 the
Step 24: Text IR is superior
Step 25: Else
Step 26: Purity is higher for effectual text IR
Step 27: End
Step 28: Return best IR text
Step 29: Compute Accuracy
Step 30: end

d. Feature reduction for IR

Here, ant-colony-based root stemmer is used for recognizing root of text after text grouping, which is similar to pattern matching approach [24]. Text roots are hauled out after eliminating prefixes and suffixes of provided word. With the use of Ant colony as stemming algorithm for feature reduction as in fig. 2, it reduces sum of features with extracted Pheromone as lexical forms of text from various sources similar to stem. This approach diminishes text document vector size and enhances learning speed and provide classification phase for numerous classifiers. Ant colony based light stemming specifies the process of cutting of lesser set of suffixes and prefixes without attempting to handle infixes or to recognize features for finding roots [25]. However, light stem has ability to appropriately conflate numerous variants of texts to larger stem classes, but still it is unable to handle other conflate forms that work together. When compared to MS-AC approach light stem attempts to classify performance by holding appropriate meaning for text.

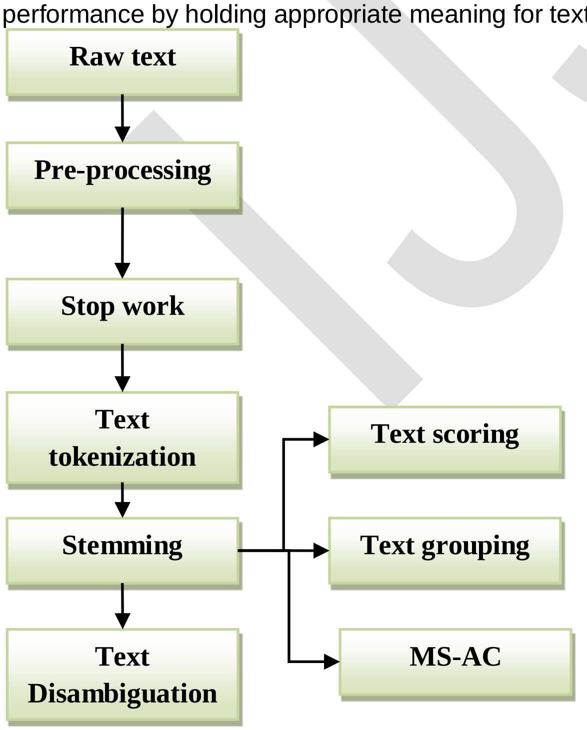

Fig 2: Flow diagram of MS-AC model

MS-AC based stemming traverse from sentence to sentence, which is computed as in Eq. (11):

$$\eta(r,u) = \frac{1}{C_u} \qquad (11)$$

Here, C_u specifies retrieval text of sentence 'u'. In parallel, Ant specifies total unvisited sentence for IR. This global updation offers effectual IR from text input.

V NUMERICAL RESULTS AND DISCUSSIONS

Based on various background revisions, investigators offer separate performance metrics to determine stemmers in two significant categories. They are: accuracy and strength of stemmers. For attained effectual solution, these two metrics are used:

a. Accuracy

Accuracy attained from stemmer depicts how these outcomes are appropriate. Like conventional recall and precision scores, accuracy is considered to be 100% only if stemmer provides all appropriate stems for all texts. Also, it does not return any added incorrect stems. If accuracy is determined to be 100%, it shows that stemmer is suitable as in fig. 3. Here, accuracy of stemmer is depicted with Equation expressed below as in Eq. (12):

$$Accuracy = \frac{TP}{TP+FP+FN} \qquad (12)$$

Where TP is total amount of suitable stemmer, FP is total amount of incorrect stems, FN is total amount of suitable stems that are not returned from stemmer as in table I.

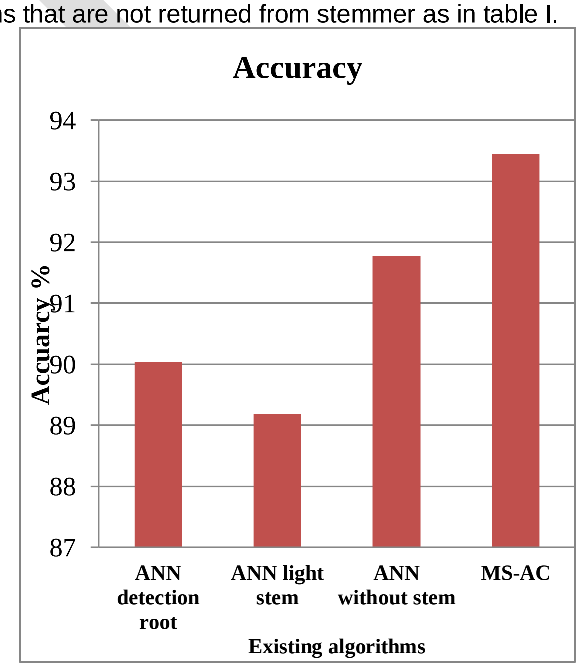

Fig 3: Accuracy computation

b. Text per conflation (TC)

Text per conflation is depicted as average amount of texts that matches similar stem. For instance, if text "information", "retrieval" and "query" are stemmed to same root, then TC score is computed as three as in Fig. 4. Value of total amount of texts per conflation projects the strength of stemmer. If stemmer value is higher, then strength is also higher. TC metric is evaluated with Equation given below Eq. (13):

$$TC = \frac{C}{S} \qquad (13)$$

Where ‘c’ specifies total amount of suitable corpus texts before stemming or total amount of word types: in a similar way, ‘s’ specifies total amount of stems returned by stemmer.

Text1: stem 1, [stem2, stem3...]
Text2: stem 1, [stem2, stem3...]
Text3: stem 1, [stem2, stem3...]
Text4: stem 1, [stem2, stem3...]
Text5: stem 1, [stem2, stem3...]

		Predicted	
		Positive	**Negative**
Actual	**Positive**	**TP**	**FN**
	Negative	**FP**	**TN**

TABLE I: CONFUSION MATRIX

c. Index compression factor

ICF specifies average reduction in index size attained via stemming process. For instance, dataset with 10000 texts and 80,000 stems have an index compression factor with 20%. If stemmers’ strength is higher, then ICF value is also higher. It is depicted as in Eq. (14):

$$ICF = \frac{C - S}{C} \tag{14}$$

Where ‘c’ specifies total amount of word types and ‘s’ specifies number of appropriate stems attaned from stemmer.

d. Text change average (TCA)

stemmers usually leave text to be unchanged. For instance, stemmers will not modify verb content, e.g. he wrote, as it is measured as root word. Stronger stems generally modify words more frequently than weaker stemmers do to acquire appropriate stem or root. TCA is formulated as in Eq. (15):

$$TCA = \frac{C - U}{C} \tag{15}$$

Where ‘C’ specifies total amount of corpus word types and ‘U’ specifies number of unique text that has not been changed after stemming procedure.

e. Average removed texts after stemming

Stronger stem pretends to eliminate or remove characters from words to generate stem. For instance, if following words, “information”, “retrieval content”, “query” and “text” are stemmed in root, then average removed characters will be determined as $\frac{1+2+1+1}{4} = 1$ characters.

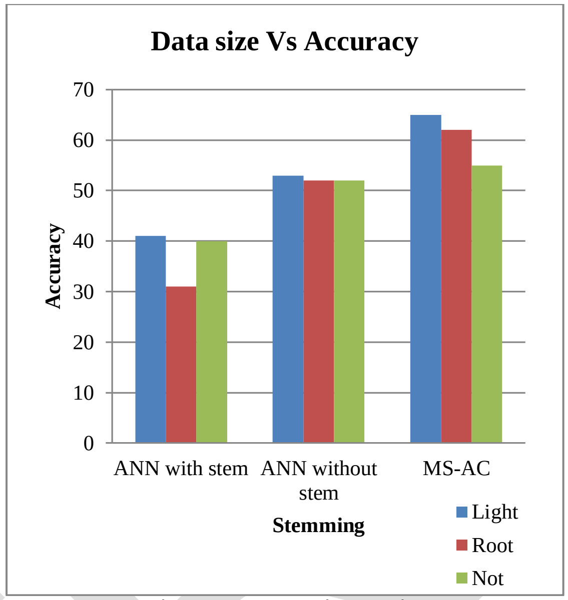

Fig 4: Input stemming results

f. Global stemming weighted score (GSWS)

This GSWS is formulated as in Eq. (15):

$$GSWS = \frac{\alpha \sum T_t}{\beta \sum Accuracy_t} \tag{15}$$

where T_w specifies time consumed by stemmer to stem text ‘t’, and $Accuracy_t$ specifies accuracy of texts attained form stemmer for text ‘T’. Certain parameters are utilized to handle weights of execution time and accuracy as in table II. Investigators may change values so as to make one element to be more specific than others. For instances, if accuracy of some investigators is set from low to high value then stemmer is effectual. Here, values are set as high, that is, 1.

TABLE II: PERFORMANCE METRICS COMPARISON

Metrics	Light stemmer	MS-AC stemmer
TP	4776	5400
FP	13570	11325
Removed character	2800	3200
Accuracy	14.96%	45.95%
TC	2.04	1.56
ICF	51.08	48.65
TCA	80.85	52.65
Execution time (s)	1.77	1.56
GSCS	0.11	0.9

MS-AC-based stemmer provided in this paper is compared with prevailing approaches to acquire better feasibility while determining benchmarking process. Benchmark computation over these stemmers is directly run as it is considered by systems.

TABLE III: IR METRICS

Measure	ANN	MS-CS
Total learning queries	200	200
Total test queries	50	50
Total cross media	4365	4379
Total relevant cross media	85	86
Total retrieved cross media	81	79
Precision after retrieving text	0.5574	0.6378

For validating and benchmarking new stemmer, investigators have two probabilities: corpse is called to be usual devoid of any modifications, next is, if investigators wishes to integrate

IJSTR©2020
www.ijstr.org

stemmer with benchmark dataset, they have to offer simple text file with outcomes of stemmer results as in Table III. These outcomes will be compared with other stem outcomes.

V CONCLUSION

In this investigation, a novel multi-source-based Ant colony stemming approach is used for information retrieval with the aim of retrieving text in accordance with information relevance. The anticipated MS-AC approach is to enhance the relevance of retrieved text. This MS-AC approach adjusts itself to user information requirements based on the relevance of present search outcomes in accordance with the relevance weight of retrieved text content. This technique depicts offering a restricted amount of ranked text files in reception to the provided user queries. The novel MS-AC technique is merged with appropriate mining and user profiling approach for the improvement of IR performance metrics fulfilling the requirement of investigation. This MS-AC model is formulated specifically to overcome certain restrictions such as lesser recall and precision value along with non-adaptation to the users of prevailing ranking approaches that eliminate a semantic investigation of the text itself. And content-sourced approaches like TF-IDF are considered as baseline approaches to compare MS-AC algorithm for validating ranking documents for user preferences. Here, search engines are depicted as back end prototype interaction amongst information sources and user information for document collection and various queries from the domain of users' expert systems for validating MS-AC technique. Evaluation was carried out in MATLAB environment, where a number of non-relevant and relevant texts of the host were known. A simulation of MS-AC technique illustrates performance metrics with improved accuracy over the selected conventional algorithms. The anticipated algorithms have certain interesting features like adaptability and scalability. Scalability shows that this approach can be merged with other approaches to adapt to various environments. This adaptation is based on user information that is essential in diverse environment. In future, this work can be extended by analyzing ranking algorithm with respect to Meta-heuristic approaches.

REFERENCES

[1] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (Eds.), Machine learning: An artificial intelligence approach. Springer Science and Business Media, 2013.

[2] Kotsiantis, S. B., Zaharakis, I., and Pintelas, P., Supervised machine learning: A review of classification techniques, 2007.

[3] Bijalwan, V., Kumar, V., Kumari, P., and Pascual, J., KNN based machine learning approach for text and document mining. International Journal of Database Theory and Application, 7(1), 61-70, 2014.

[4] F. Harrag, and E. El-Qawasmeh, "Neural Network for Arabic Text Classification", Proceedings of ICADIWT'09, pp. 778-783, 2009.

[5] R.M. Duwairi, M.N. Al-Refai, N. Khasawneh, "Feature Reduction Techniques for Arabic Text Categorization", Journal of the American society for information science and technology, 60(11), pp. 2347-2352, 2009.

[6] M. N. Singh, "The Improved K-Means with Particle Swarm Optimization," Journal of Information Engineering and Applications, Vol.3, No.11, 2013.

[7] O. Ghanem and W. Ashour, "Stemming Effectiveness in Clustering of Arabic Documents," International Journal of Computer Applications (0975 – 8887), 2012.

[8] S. Alghamdi and Shahriza., "Arabic web pages clustering and annotation using semantic class features," Journal of King Saud University, Computer and Information Sciences, 2014.

[9] Kelaiaia and H. Merouani, "Clustering with Probabilistic Topic Models on Arabic Texts: A Comparative Study of LDA and K-Means," The International Arab Journal of Information
Technology VOL. 13, NO. 2, 2016.

[10] Ismael and N. Hashimah, "Text Document Preprocessing and Dimension Reduction Techniques for Text Document Clustering," International Conference on Artificial Intelligence with Applications in Engineering and Technology, 2014.

[11] M.M. Syiam, Z.T. Fayed, and M.B. Habib, "An Intelligent System for Arabic Text Categorization", IJICIS, 6(1), pp. 1-19, 2006.

[12] A. A. Mesleh, "Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System", Journal of Computer Science, 3(6), pp. 430-435, 2007.

[13] R.M. Duwairi, M.N. Al-Refai, N. Khasawneh, "Feature Reduction Techniques for Arabic Text Categorization", Journal of the American society for information science and technology, 60(11), pp. 2347-2352, 2009.

[14] F. Harrag, and E. El-Qawasmeh, "Neural Network for Arabic Text Classification", Proceedings of ICADIWT'09, pp. 778-783, 2009.

[15] Rajendran T et al, "Recent Innovations in Soft Computing Applications", Current Signal Transduction Therapy, Vol. 14, No. 2, pp. 129 – 130, 2019.

[16] Emayavaramban G et al, "Indentifying User Suitability in sEMG based Hand Prosthesis for using Neural Networks", Current Signal Transduction Therapy, Vol. 14, No. 2, pp. 158 – 164, 2019.

[17] Rajendran T & Sridhar K P, "Epileptic seizure classification using feed forward neural network based on parametric features". International Journal of Pharmaceutical Research, 10(4): 189-196, 2018.

[18] Hariraj V et al, "Fuzzy multi-layer SVM classification of breast cancer mammogram images", International Journal of Mechanical Engineering and Technology, Vol. 9, No.8, pp. 1281-1299, 2018.

[19] Muthu F et al, "Design of CMOS 8-bit parallel adder energy efficient structure using SR-CPL logic style", Pakistan Journal of Biotechnology, Vol. 14, No. Special Issue II, pp. 257-260, 2017.

[20] Yuvaraj P et al, "Design of 4-bit multiplexer using Sub-ThresholdAdiabatic Logic (STAL)", Pakistan Journal of Biotechnology, Vol. 14,No. Special Issue II, pp. 261-264, 2017.

[21] Keerthivasan S et al, "Design of low intricate 10-bit current steering digital to analog converter circuitry using full swing GDI", Pakistan Journal of Biotechnology, Vol. 14, No. Special Issue II, pp. 204-208, 2017.

[22] Vijayakumar P et al, "Efficient implementation of decoder using modified soft decoding algorithm in Golay (24, 12) code", Pakistan Journal of Biotechnology, Vol. 14, No. Special Issue II, pp. 200-203, 2017.

[23] Rajendran T & Sridhar K P, "Epileptic Seizure-Classification using Probabilistic Neural Network based

on Parametric Features", International Journal of Scientific & Technological Research, Vol.9, No. 3, 2020 (Accepted for Publication).

[24] Rajendran T et al, "Performance analysis of fuzzy multilayer support vector machine for epileptic seizure disorder classification using auto regression features", Open Biomedical Engineering Journal, Vol. 13, pp. 103-113, 2019.

[25] Rajendran T et al, "Advanced algorithms for medical image processing", Open Biomedical Engineering Journal, Vol. 13, 102, 2019.

[26] Anitha T et al, "Brain-computer interface for persons with motor disabilities - A review", Open Biomedical Engineering Journal, Vol. 13, pp. 127-133, 2019.