Jurnal Teknologi

Full Paper

COMPARISON OF VSM, GVSM, AND LSI IN INFORMATION RETRIEVAL FOR INDONESIAN TEXT

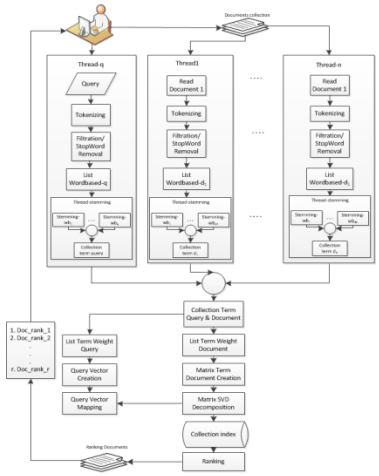
Jasman Pardede*, Milda Gustiana Husada

Informatics Department, Faculty of Industrial Technology, Institut Teknologi Nasional (Itenas), Bandung, Indonesia

*Corresponding author
jasman@itenas.ac.id

Article history
Received
5 June 2015
Received in revised form
7 September 2015
Accepted
19 December 2015

Graphical abstract



Abstract

Vector space model (VSM) is an Information Retrieval (IR) system model that represents query and documents as ***n***-dimension vector. GVSM is an expansion from VSM that represents the documents base on similarity value between query and minterm vector space of documents collection. Minterm vector is defined by the term in query. Therefore, in retrieving a document can be done base on word meaning inside the query. On the contrary, a document can consist the same information semantically. LSI is a method implemented in IR system to retrieve document base on overall meaning of users' query input from a document, not based on each word translation. LSI uses a matrix algebra technique namely Singular Value Decomposition (SVD). This study discusses the performance of VSM, GVSM and LSI that are implemented on IR to retrieve Indonesian sentences document of .pdf, .doc and .docx extension type files, by using Nazief and Adriani stemming algorithm. Each method implemented either by thread or no-thread. Thread is implemented in preprocessing process in reading each document from document collection and stemming process either for query or documents. The quality of information retrieval performance is evaluated based-on time response, values of recall, precision, and F-measure were measured. The results show that for each method, the fastest execution time is .docx extension type file followed by .doc and .pdf. For the same document collection, the results show that time response for LSI is more faster, followed by GVSM then VSM. The average of recall value for VSM, GVSM and LSI are 82.86 %, 89.68 % and 84.93 % respectively. The average of precision value for VSM, GVSM and LSI are 64.08 %, 67.51 % and 62.08 % respectively. The average of F-measure value for VSM, GVSM and LSI are 71.95 %, 76.63 % and 71.02 % respectively. Implementation of multithread for preprocessing for VSM, GVSM, and LSI can increase average time response required is about 30.422%, 26.282%, and 31.821% respectively.

Keywords: VSM, GVSM, LSI, performance, multithread

1.0 INTRODUCTION

Information Retrieval (IR) is a method to retrieve unstructured data stored in a documents collection and be able to provide the information required according to query input user [1-5]. IR system aims to retrieve the relevant documents and at the same time to retrieve the irrelevant documents minimally [5, 6].

Vector Space Model (VSM) is a model of conventional IR which represents a document collection into a form of term-document matrix. VSM considers the document as a vector in a high dimension space. The similarities between the input query and the document collection is measured by vector cosine [7, 8]. VSM assumes that the query and the document terms as linearly independent. Generalized Vector Space Model (GVSM) overcomes the VSM by changing the terms as linearly independent to be terms that have correlation to another term. By using the correlation matrix, GVSM reduces the error expressed in VSM by assuming term independence [8-11]. The term-document matrix has a high dimension space and

independent in each term, therefore it is highly vulnerable to noise. To solve this problem by reducing the matrix dimension. Latent Semantic Indexing (LSI) improves GVSM model by reducing the space dimension of term-document matrix. The derived LSI-space features are orthogonal and bring observation data variance that using relatively smaller dimension [12-15].

This study discussed the performance of each method that presented base on examination results of precision, recall, F-measure and time response. Examined documents are Indonesian language documents which are formated as docx, doc and pdf. Each method has been observed in the effect of either with thread or no-thread implementation for time response parameter. Thread is a process that running in a program or operating system. Multithreading is the ability of a program or operating system to execute and handle user requests more than one user without duplication or copying program that runs on a computer at the same time [16].

This paper will be presented as follows. Section 2 to Section 4 explains about VSM, GVSM and LSI respectively. Section 5 discusses research methodology, which is followed about analysis and results in Section6. Finally, the conclusion in Section 7.

2.0 VECTOR SPACE MODEL

VSM is a model of the IR system that represents each query and the document as a vector of ***n*** -dimension. In VSM, each document in the document collection and query are represented by a multi-dimensional vector [7, 8]. Each dimension of the vector is represented by a single term. The term that is used usually bases on the term in query or keyword, so that the term in document collection but not in the queries are usually be ignored. Each document is represented by a weight vector $\vec{d} = (w_{1j}, w_{2j}, \dots, w_{tj})^T$, where w_{ij} is the weight of *i*-th term in document of d_j, t is the number of terms in the document collection. Document collection $A = \vec{d}$ is represented by a term-document matrix with t rows and d columns. The query vector is represented by $\overrightarrow{q_k} = (q_{1j}, q_{2j}, \dots, q_{tj})^T$ where q_{ij} is weight of z-th term in $\overrightarrow{q_k}$ query. The similarity between the document and the query is determined by ***sim*** (d_j, q), as follows:

$$sim(d_j, q) = \frac{d_j * q}{\|d_j\| \|q\|} = \frac{\sum_{i=1}^{N} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \quad (1)$$

In VSM, terms either in the documents or given keywords, do not show semantic meaning similarity which is contained in document and query, however VSM decreases the performance in case of synonym and polysemy problems. Synonym may decrease recall value, otherwise polysemy may decrease precision value [9, 15].

3.0 GENERALIZED VECTOR SPACE MODEL

GVSM is an extension from the VSM by adding the type of additional information to retrieve document. IR by using GVSM represents document with vector similarity of document collection [9-11].

In 1985, Wong *et al*., proposed GVSM as an alternative of IR VSM. The VSM assumes the term as linearly independent while GVSM avoids that assumption by using documents as vector space than term. GVSM modifies the VSM by introducing ad hoc schemes, namely the inter-term relationship or correlation. The correlation matrix provides a relational model which is obtained from the document collection. The correlation between the two indexes of term depends on the number of documents in two terms that appears simultaneously. Suppose A_{txd} is the term-document matrix, then GVSM calculates the term correlation matrix R_{txt} by multiplying A with its transpose, A^T. The similarity query term vector and a document matrix are determined by dot product between the query vector, matrix and correlation matrix term document, that stated by ***sim*** (d_j, q), as follows:

$$sim(d_j, q) = \frac{\sum_{j=1}^{n} \sum_{i=1}^{n} w_{i,j} * w_{j,q} * t_i . t_j}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} * \sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \quad (2)$$

where t_i and t_j are term vectors in a 2-dimensions vector space; d_j and q are document vector and query vector respectively, w_{ij} is the weight of term-document; w_{iq} is the weight of query's term; and n is the dimension of space.

4.0 LATENT SEMANTIC INDEXING

Latent Semantic Indexing (LSI) is a method that implemented in the IR system to retrieve information based on the overall meaning of a document not just only by the meaning of word per word. LSI is a variant of VSM that maps a high dimension space into a lower dimensional space [12-15].

LSI is a technique that projecting queries and documents into a "latent" semantic dimension space. In the late nt semantic space, the query and the document may have a high similarity value eventhough the terms that owned by the same document does not have the same literally meaning, but in polysemy or synonym they contain the same meaning.

Suppose the weight of query and document term is A_{txd} matrix. To reduce the dimension of document matrix, LSI uses algebra techniques, namely *Singular Value Decomposition* (SVD) [15]. By SVD, A matrix will be decomposed into three matrices namely $T_{t,n}$, $S_{n,n}$, $D_{d,n,}$, then $A_{t,d} = T_{t,n}.S_{n,n}.(D_{d,n})^T$ where t is the number of terms, d is the number of documents, and $n = min\ (t,d)$. T and D matrices are orthonormal columns, so that $TT^T = D^TD = I$, $rank\ (A) = r$, $S = diag\ (\sigma_1, \sigma_2, \dots, \sigma_n)$, $\sigma_i > 0$ for $1 \le i \le r$, $\sigma_j = 0$ for $j \ge r+1$. The query, which is processed as document,

is also a set of terms that can be represented as a vector in a k-dimension space, and then are compared with the documents. So that the user queries can be expressed as $\hat{q} = q^T T_{t,k} S_{k,k}^{-1}$. The similarity of document matrix and query vector is defined by ***sim (d$_j$, q)***, as follows:

$$sim(\overrightarrow{D_j}, \vec{\hat{q}}) = \frac{\overrightarrow{D_j} * \vec{\hat{q}}}{|\overrightarrow{D_j}| * |\vec{\hat{q}}|} \quad (3)$$

5.0 METHODOLOGY

This study concerns about performance of VS M, GVSM and LSI methods in time response, precision, recall and F-measure when processing Indonesian language documents. Weight of term is defined at preprocessing stage which uses Adriani-Nazif stemming algorithm that more suitable for Indonesian language stemming process [17, 18]. All of the methods applied with thread and no-thread.

The use of thread on each method is performed at the preprocessing stage, i.e. reading each document from document collection and stemming process, both on the query and the word-based that contained in a document. Thread creation at document reading is based on the number of documents held by the document collection while the thread creation at stemming process is carried out based on the number of word-based contained in the list of document. Each document is given a thread to produce a list of terms that are owned by the document. Each of lists of document terms will be processed by tokenizing and stop word filtration to provide word-based list. A thread stemming is conducted in each the word-based list to provide root-word. The result of thread in stemming process is gathered to get the list root word of a document. A root word in the user input query is obtained by using the same process that based on the word-based list from query. Weight of term resulted by the user input query stemming process are gathered with the results from documents stemming in the document collection. The unification of query and document weight of terms will create the document matrix. The relevant documents which retrieved by the system is determined by VSM, GVSM, and LSI. Multithread implementation scheme in the LSI method is as described on Figure 1.

While the IR which no-multithread implementation is obtained by the sequences process for both in the reading of the document file and the stemming process. Therefore, in determining the word-based list that contained in other documents may only be done after the word-based list of document has been provided. Whilst the stemming of each word-based list that in a document can be obtained after the root word of the word-based list has already been produced.

In LSI method, to develop SVD from document matrix is carried out by using Efficient Java Matrix Library (EJML) API function while the VSM and GVSM are not.

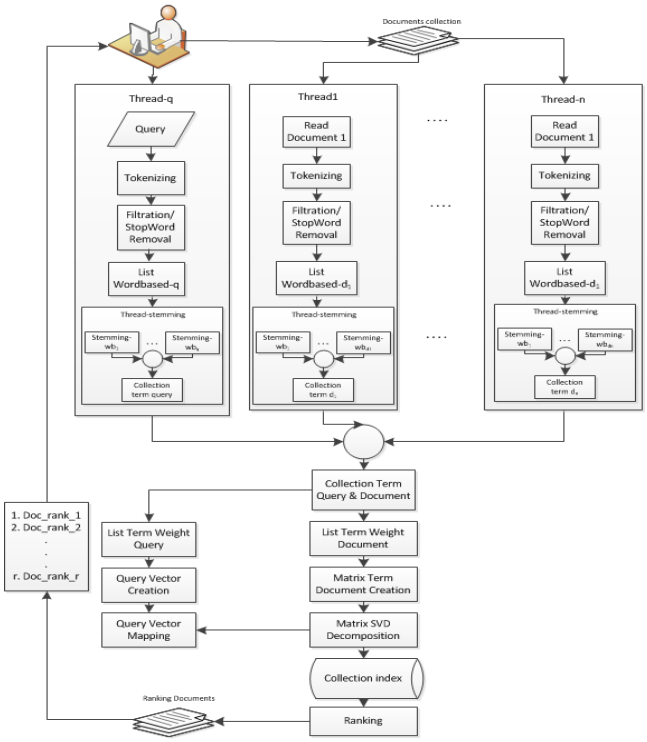


Figure 1 IR LSI scheme that uses thread

6.0 EXPERIMENTAL ANALYSIS

There are four parameters used to measure the performance of IR system, i.e., the response time, recall, precision, and F-measure [19-22]. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage. Presicion is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage. F-measure is a simple measuring tool for comparing recall and precision. Refer to [19], the default value of balanced F-measure are $\alpha = ½$ and $\beta = 1$. This study applied $\alpha = ½$ and $\beta = 1$. Table 1 summarizes characteristics of the datasets that have been applied.

The success or failure of an IR method is very dependent on the term weighting schemes [21]. In this study, the term weighting scheme of VSM, GVSM, and LSI method are equal. The evaluation to docx, doc and pdf types of Indonesian language document collection was conducted to provide the value of precision, recall, F-measure and time response. Document collection has 10 groups of folder that consist of 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 documents which have identical format. Each of these documents was examined to docx, doc, and pdf document format. To read a document in Microsoft Word, the Apache Point function was used. The docx format used XML Beans function, the doc format used DOM4J function and the pdf format used PDFBox function.

The evaluation of precision, recall, and F-measure for each method are shown by Table 1. The precision value of VSM, GVSM, and LSI always more than 50%. The average precision value of VSM is 64.08%, GVSM is 67.51%, and LSI is 62.08%. The recall value of VSM, GVSM, and LSI always greater than 70.59%. The average of recall value of VSM is 82.86%, GVSM is 89.68%, LSI is 84.93%. For number of documents are 10, the result of precision and recall of VSM and GVSM are 100% and LSI is 80%. For number of documents are below 50 then the result of precision and recall from higher to lower are GVSM, VSM and LSI consecutively.

When the number of documents are more than 80 then the result of precision and recall from higher to lower are LSI, GVSM and VSM consecutively. This shows that the number of document in a document collection increases then LSI method yields the better performance. The result of F-measure for VSM, GVSM, and LSI method are more than 58.54% with the best average number is for GVSM method.

The 10 groups of folder were examined to evaluate VSM, GVSM, and LSI method that are implemented either thread or no-thread in respect of time response required. The result of time response of each method for doc format documents is shown numerically by Table 2 and graphically by Figure 2. Figure 2 shows that the time response LSI is the fastest. However, by implementing a thread into GVSM method, its execution time is faster than LSI method that no thread applied. As shown by Figure 3 to Figure 5, with thread and no-thread implementation to each method (VSM, GVSM and LSI) for execution time response have significant effect especially when the number of term increases, i.e. when the number of document collection is 100.

Time response of docx, doc, and pdf format document were determined by test to each document collection. The time response of GVSM which implemented no-thread for each format is given by Table 3. This shows that docx format is the fastest then followed by doc, and pdf format.

Table 1 The result of precision, recall, F-measure of VSM, GVSM, and LSI method

No	Number of documents	Precision			Recall			F-Measure		
		VSM	GVSM	LSI	VSM	GVSM	LSI	VSM	GVSM	LSI
1	10	100.00	100.00	80.00	100.00	100.00	66.67	100.00	100.00	72.73
2	20	78.57	78.57	71.43	91.67	91.67	83.33	84.62	84.62	76.92
3	30	73.33	73.33	62.50	91.67	91.67	83.33	81.48	81.48	71.43
4	40	64.71	70.59	58.82	91.67	100.00	83.33	75.86	82.76	68.97
5	50	55.00	60.00	52.38	78.57	85.71	78.57	64.71	70.59	62.86
6	60	60.00	61.90	63.64	70.59	76.47	82.35	64.86	68.42	71.79
7	70	50.00	60.00	55.56	70.59	88.24	88.24	58.54	71.43	68.18
8	80	55.17	58.06	59.38	80.00	90.00	95.00	65.31	70.59	73.08
9	90	54.05	57.89	61.54	76.92	84.62	92.31	63.49	68.75	73.85
10	100	50.00	54.76	55.56	76.92	88.46	96.15	60.61	67.65	70.42
Average		**64.08**	**67.51**	**62.08**	**82.86**	**89.68**	**84.93**	**71.95**	**76.63**	**71.02**

Table 2 Time response evaluation of VSM, GVSM, and LSI method that applied thread or no-thread to doc format

No	Number of documents	Non Thread (s)			Thread (s)			Increasing by thread (%)		
		VSM	GVSM	LSI	VSM	GVSM	LSI	VSM	GVSM	LSI
1	10	5.319	5.874	4.353	4.035	4.336	3.572	24.140	26.183	17.942
2	20	7.11	8.28	6.349	5.788	5.694	4.852	18.594	31.232	23.579
3	30	11.02	11.31	10.621	10.014	10.827	9.064	9.129	4.272	14.660
4	40	22.652	23.474	15.647	12.497	14.714	11.138	44.830	37.318	28.817
5	50	28.002	29.273	20.264	15.537	18.24	13.791	44.515	37.690	31.943
6	60	34.42	36.366	30.779	21.515	22.901	21.497	37.492	37.026	30.160
7	70	37.146	39.486	36.551	24.445	27.838	25.132	34.192	29.499	31.241
8	80	44.707	46.637	40.13	31.674	32.424	30.311	29.152	30.476	24.470
9	90	50.844	53.216	47.425	33.108	35.61	32.198	34.883	33.084	32.108
10	100	61.693	64.297	50.653	37.702	40.219	36.52	38.887	37.448	27.902
	Average	31.821	26.277	19.632	21.281	18.8075	31.582	30.422	26.283	31.821

Table 3 Time response of GVSM method to docx, doc, and pdf format (when no thread applied)

No	Number of documents	Non Thread (s)		
		docx	doc	pdf
1	10	5.242	5.874	6.864
2	20	7.815	8.280	9.470
3	30	11.014	11.31	13.281
4	40	22.852	23.474	30.868
5	50	28.443	29.273	34.15
6	60	35.822	36.366	42.966
7	70	38.142	39.486	46.228
8	80	44.984	46.637	52.944
9	90	52.456	53.216	61.887
10	100	62.479	64.297	74.066

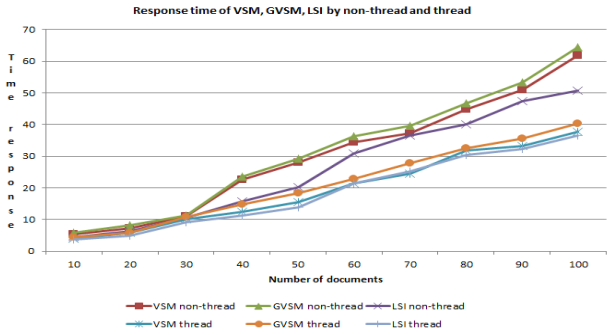


Figure 2 Graph of time response of VSM, GVSM, and LSI using thread vs. non-thread

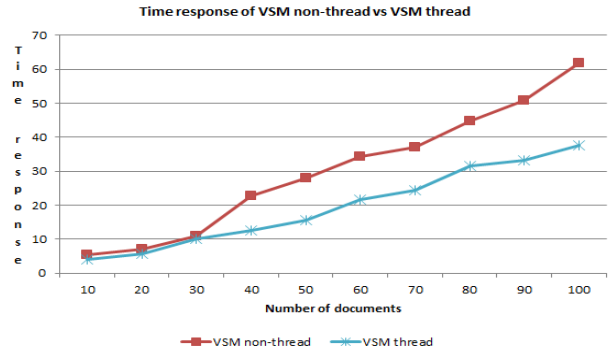


Figure 3 Graph of time response using thread vs. non-thread of VSM

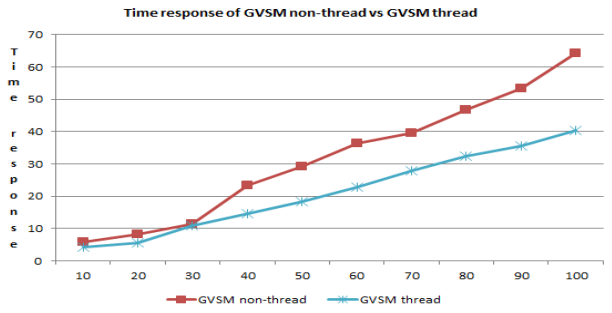


Figure 4 Graph of time response using thread vs. non-thread of GVSM

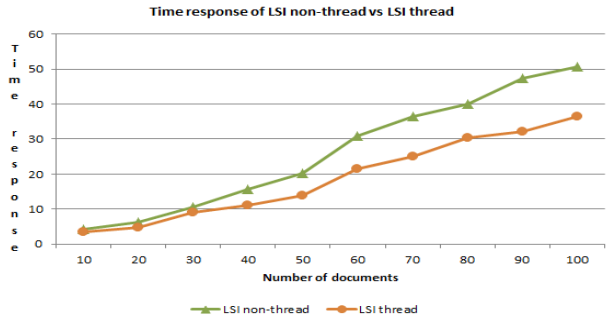


Figure 5 Graph of time response using thread vs. non-thread of LSI

7.0 CONCLUSION

Based on the result for time response, the value of precision, recall and F-measure showed that GVSM has better value of precision and recall compared to VSM. Yet the time response of VSM method always less than GVSM method. Based on increasing number of documents, the precision and recall showed that LSI has better result than GVSM or VSM. In case of F-measure value, for smaller number of documents, LSI has a smaller value than VSM and GVSM, but for higher number of documents LSI more better than GVSM and VSM. Time response of LSI is more faster than VSM and GVSM. The use of thread in VSM method is more effective than GVSM and LSI.

References

[1] Goker, A., and Davies, J. 2009. *Information Retrieval: Searching In The 21st Century*. United Kingdom: A John Wiley and Sons, Ltd., Publication.

[2] Ingwersen, I. and Järvelin, K. 2005. *The Turn: Integration Of Information Seeking And Retrieval In Context*. Springer.

[3] Kowalski, G. 1997. *Information Retrieval System Theory and Implementation*. United States of America: Kluwer Academic Publishers,

[4] Manning, C., D., *et al*. 2009. *An Introduction to Information Retrieval*. England: Cambridge University Press.

[5] Yates, R. B., and Neto, B. R. 1999. *Modern Information Retrieval.* New York: ACM Press.

[6] Robertson S. 2007. *On Document Populations and Measures of IR Effectiveness*, UK: Microsoft Research, Cambridge.

[7] Berry, M. W., Drmac, Z. and Jessup, E. R. 1991. Matrices, Vector Spaces, and Information Retrieval. *SIAM Review*.

[8] Signh, N. J., and Dwivedi, S. K. 2012. Analysis of Vector Space Model in Information Retrieval. *International Journal of Computer Applications*.

[9] Wong, S. K. M., Ziarko, W., and Wong, C. N. P. 1985. Generalized Vector Space Model in Information Retrieval. *Proceedings of 8th ACM SIGIR Conference on Research and Development in Information Retrieval*.

[10] Soboroff, I., and Nicholas, C. 2000. Collaborative Filtering and The Generalized Vector Space Model. Athen, Greece. 351-353.

[11] Tsatsaronis, G., and Panagiotopoulou, V., 2009. A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness, *Proceedings of the EACL 2009 Student Research Workshop*. Athen, Greece. 70-78.

[12] Deshmukh, A., and Hegde, G. 2012. A Literature Survey On Latent Semantic Indexing, *International Journal of Engineering Inventions*. 1(4): 2278-7461.
[13] Kontosthathis, A., and Pottenger, W. M. 2006. A Framework for Understanding Latent Semantic Indexing Performance. *Journal of Information Processing and Management*.
[14] Rosario, B. 2000. *Latent Semantic Indexing: An Overview*. INFOSYS 240, Spring 2000.
[15] Waraich, N. K., Sinder, H. S. 2014. Text Search Optimization Using Latent Semantic Indexing, *International Journal of Computer Science and Information Technologies*. 5(5): 0975-9646.
[16] Oracle. 2012. *Multithreaded Programming Guide*. Oracle And/Or Its Affiliates.
[17] Nazief, B., Adriani, M. Confix-Stripping: Approach to Stemming Algorithm for Bahasa Indonesia. Faculty of Computer Science University of Indonesia.
[18] Asian, J., Williams, H. E., and Tahaghoghi, S. M. M. 2005. Stemming Indonesia, Australia: School of Computer Science and Information Technology.
[19] Power, D. M. W. 2011. Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*. 37-63.
[20] Smucker, M. D, and Clarke, C. L. A. 2012. Time-Based Calibration of Effectiveness Measures, *SIGIR'12*. Portland, Oregon, USA.
[21] Jensi, R., and Jiji, W.G. 2013. A Survey on Optimization Approaches to Text Document Clustering, *International Journal of Computational Sciences & Applications*. 3(6): 10.5121.
[22] Datta, J., and Bhatttacharyya, P. 2010. *Ranking Information Retrieval.*