Journal of Information Systems and Informatics

Vol. 1, No. 2, September 2019 e-ISSN: 2656-4882 p-ISSN: 2656-5935

http://journal-isi.org/index.php/isi Published By DRPM-UBD

Analysis of Document Clustering based on Cosine Similarity and K-Main Algorithms

Bambang Krismono Triwijoyo[1], Kartarina[2]

[1,3]Computer Science Departement, Bumigora University, Mataram, Indonesia
[2]Information Technology Departement, Bumigora University, Mataram, Indonesia
Email: [1]bkrismono@universitasbumigora.ac.id, [2]kartarina@universitasbumigora.ac.id

Abstract

Clustering is a useful technique that organizes a large number of non-sequential text documents into a small number of clusters that are meaningful and coherent. Effective and efficient organization of documents is needed, making it easy for intuitive and informative tracking mechanisms. In this paper, we proposed clustering documents using cosine similarity and k-main. The experimental results show that based on the experimental results the accuracy of our method is 84.3%.

Keywords: document clustering, cosine similarity, k-main

1. INTRODUCTION

Document clustering has become an increasingly important technique for the organization of documents without supervision, automatic topic extraction, and rapid retrieval of information. Categorizing electronic documents such as scientific papers based on certain topics requires extra time and effort, making it difficult for users to categorize relevant documents. The clustering method can be used to automatically group documents according to specific topics. In this case, the efficiency of grouping is highly desirable because of the requirements of high data volumes. The clustering algorithm is mainly categorized into Hierarchical and Partitioning methods. The hierarchical grouping method works by grouping data objects into cluster trees, while the partitioning method groups documents based on partition [1].

In this paper we use the K-means method for document clustering into 3 topic categories, where after documents are converted to plain text format, then pre-processing is carried out, among others, tokenize, stop-words filtering, stemming and conversion of all characters into lowercase letters, then mapped to a high dimensional vector with one dimension per "term". We use cosine similarity to measure the similarities between the two vectors. After this introduction the second part will be discussed our methods about clustering documents, followed

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

by the third part the results and discussion are described. In the end, we explained the conclusion and our future work.

Text documents are represented as a set of words, where words are assumed to appear independently and not sequentially. The "bag of word" model is widely used in information search and text mining [2]. Words are counted in a bag, which is different from the mathematical definition of a set. Each word is related to dimensions in the data space that is generated and each document then becomes a vector consisting of non-negative values in each dimension. Here we use the frequency of each term as its weight, which means the terms that appear more often are more important and descriptive for documents.
Let $D= \{d_1,\ldots,d_n\}$ be a set of documents and $T = \{t_1,\ldots, t_m\}$ the set of distinct terms occurring in D. A document is represented as an m-dimensional vector $\overrightarrow{td}$. Let *tf (d,t)* denote the frequency of term $t \in T$ in document $d \in D$. Then the vector representation of document d is

$$\overrightarrow{td} = (tf(d,t_1), \ldots, tf(d,t_m)) \quad (1)$$

Although words that are more often considered more important, this usually does not occur in practice. For example, words such as "a" and "the" may be the words that appear most often in English text, but both are not descriptive or important for the subject of the document.

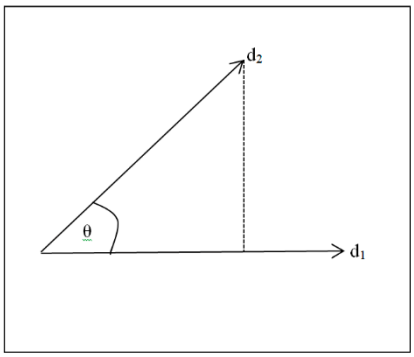

Figure 1. Angles in two-dimensional space

More complex strategies such as the Inverse document frequency (*idf*) weighting scheme are widely used. Documents are presented as vectors and measured by the level of similarity between two documents as a correlation between the vectors, which can then be quantified as angular cosines between the two vectors. Figure 1 show angles in two-dimensional space but in practice document space usually have tens or even thousands of dimensions.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The term is basically words with some standard transformations in vector representations of basic terms. First, it removes stop words or non-descriptive words for document topics, such as "a", "and", "or" and "do". In English text documents, there are approximately 527 stop words.

Second, words were stemmed using the Porter algorithm [3], so words with different endings will be mapped into one word. For instance, production, produce, produces and product will be mapped to the stem product. The underlying assumption is that different morphological variations of word with the same root/stem are thematically similar and should be treated as a single word. Third, Elimination of words that appear with a threshold frequency less than a limit, because in many cases they are not too descriptive of the subject of the document and contribute little to the similarity between the two documents. Rare terms can also be removed from the grouping process and make similarity calculations more efficient.

The clustering process is compared to similarities between two groups or between groups and objects. In hierarchical clustering, this is usually calculated as a complete link, single link or average link distance [4]. However, in the partition clustering algorithm, a cluster is usually represented by a centroid object. For example, in the K-means algorithm, the centroid of a cluster is the average of all objects in the cluster, the centroid value in each dimension is the arithmetic average of the dimensions above all objects in the cluster. Let C be a set of documents. The centroid is defined as:

$$\overrightarrow{tc} = \frac{1}{|C|} \sum_{\overrightarrow{td} \in C} \overrightarrow{td} \quad (2)$$

Where the average value of all term vectors at the set. Then vector normalization, the most frequently occurring terms are not always the most informative. Conversely, terms that often appear in a small number of documents but rarely in other documents tend to be more relevant and specific for certain groups of documents, and therefore more useful for finding similar documents. Term frequency *tf(d,t)* and the weighting scheme *tf.idf* (term frequency and inverse document frequency) is the frequency weight of a term t in document d with a factor that ignores its importance with its appearance in the entire document, which is defined as:

$$tf.idf(d,t) = tf(d,t) \times \log(|D|\, df(t)) \quad (3)$$

Where df (t) is the number of documents where the term appears, d is the document and t is the term.
Clustering, in general, is an important and useful technique that automatically organizes the collection of a large number of data objects into a small number of

coherent groups [4-5]. In certain text documents, clustering has proven to be an effective approach and is widely applied in several search engines to help identify users quickly and focus on relevant sets of results, as well as to provide collaborative recommendations. In bookmarks or collaborative tagging, a group of users who share certain characteristics identified from their annotations.

Many grouping methods have been proposed, such as k-means [6], naive Bayes or Gaussian models [7-9], single links [7] and DBSCAN [10]. From a different perspective, this grouping method can be classified into agglomerative or divisive, hard or fuzzy, deterministic or stochastic. The task of clustering documents has very high dimensions, ranging from several hundred to thousands of dimensions, so first need to project documents into lower-dimensional subspaces where the document semantic structure becomes clear. In low dimensional semantic space, traditional grouping algorithms can be applied. For this purpose, spectral clustering [11-12], clustering using LSI [13] and clustering based on non-negative matrix factorization [14-15] are the most commonly used techniques.

Text document clustering groups similar documents that form coherent clusters, while different documents are separated into different clusters. However, the definition of pairs of documents that are similar or different is not always clear and usually varies with setting the actual problem. For example, when grouping research papers, two documents are considered the same if they share the same thematic topic. This type of grouping can be useful for further analysis and use of datasets such as information retrieval and information extraction, by grouping the same types of information sources together. Accurate grouping requires a precise definition of the closeness between a pair of objects, in similarity or distance. Various similarities or distance measurements have been proposed and widely applied, such as the cosine similarity and Jaccard correlation coefficient. The similarity calculation between documents is measured using a simple matching coefficient [16] and the Vector Space Model method in determining the similarity percentage of each document [17]. Meanwhile, similarities are often conceived in terms of inequality or distance as well [18]. Steps such as Euclidean distance and relative entropy have been applied in grouping to calculate the distance of the object pair.

Spectral clustering shows its ability to handle highly non-linear data (data space has high curvature in each local area). Also, strong connections to differential geometry make it able to find document space type structures. Spectral grouping usually groups data points using the top eigenvector of the Laplacian graph, which is defined in the data point affinity matrix. Spectral clustering tries to find the best chart pieces so that the function of the predetermined criteria can be optimized. Many criteria functions, such as cutting ratios [19], average

associations [11], normalized pieces [11], and minimum pieces [9] have been proposed along with related problems to find their optimal solutions. From the perspective of dimensional reduction, spectral clustering infuses data points into low-dimensional spaces where traditional grouping algorithms such as K-means can be applied. One of the main disadvantages of the spectral clustering algorithm is that they use dimensional reduction which is only defined in training data. They must use all data points to study embedding so that the data set is very large and will cause expensive computing costs, which limits the application of spectral grouping to large data sets.

Latent Semantic Indexing (LSI) [20] is one of the most popular linear document indexing methods that produce low dimensional representations. LSI aims to find the best sub-space estimates into the original document space in the sense of minimizing global reconstruction errors. In other words, LSI seeks to uncover the most representative features of the most discriminatory features for document representation. Therefore, LSI may not be optimal in distinguishing documents with different semantics which is the final goal of grouping. Xu et al. apply the Non-negative Matrix Factorization (NMF) algorithm for grouping documents [14-15]. They model each cluster as a linear combination of data points, and each data point as a linear combination of clusters. And they calculate linear coefficients by minimizing global reconstruction errors from the data points using the Non-negative Matrix Factorization. Thus, the NMF method still focuses on the global geometric structure of the document space. In addition, repetitive update methods for solving NMF problems are computationally expensive.

2. METHODS

As shown in Figure 2, in general, our method consists of seven stages of the process. The first document file was collected in a folder, in this study we used 83 paper documents with 3 kinds of topics randomly, then document files that were still in pdf format were converted to plain text format, we used the Zilla PDF to TXT converter application. The second step is tokenization. The tokenized process is an integral part of the information retrieval system, involving the pre-processing of the given document and producing each token [21].

The tokenization technique calculates tokens to set the value of "Word Count or Token Count" which can be used as an indexing/ranking process. Figure 3 shows the tokenization algorithm. The third step is filtering to remove stop-words from the document. The stop-words list or stop-lists is a list of words that do not contain information. They were first introduced in 1958 by Hans Peter

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Luhn, a computer scientist, and information expert who paved the way for automatic indexing and information retrieval. Removing stop-words from indexing can reduce the space and time needed by 30-50%. This innovation was adopted by van Rijsbergen [22] where he suggested a list of 250 stop-words in English. Since then they have formed a classic keyword list, used by default or as a basis in a text database. The fourth step is stemming. Stemming is the process of reducing words that are inflected or derived form basic words. In this study, we use the Porter stemming algorithm. Porter Stemmer is a Stemmer merger developed by Martin Porter at the University of Cambridge in 1980.

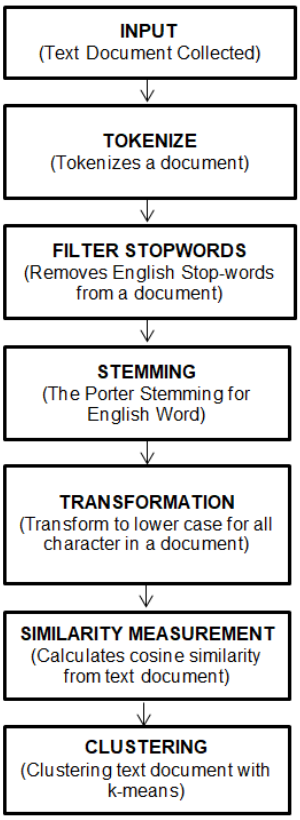

Figure 2. Document Clustering Method.

Stemmer is a sensitive end context suffix algorithm. This is the most widely used stemmer and implementation is available in many languages. But the number of definitions of stemmer need to be made before the steps can be explained. The following definitions are presented in [23]. Consonants are letters other than A, E, I, O or U and besides Y which are preceded by consonants. For examples in the word boy, consonants are B and Y, but in their experiments T and R. Vowels

are any letters that are not consonants. A consonant list greater than or equal to length one will be denoted by C and a list of vowels that are equal to V [23]. Any word can, therefore, be represented by the single form;

$$[C](VC)^m[V] \quad (4)$$

Where the m denotes m repetitions of VC and the square brackets [ ] denote the optional presence of their contents [23]. The value m is called the measure of a word and can take any value greater than or equal to zeros and is used to decide whether a given suffix should be removed. All such rules are of the form; (condition) $S_1 \rightarrow S_2$ which means that the suffix S_1 is replaced by S_2 if the remaining letters of S_1 satisfy the condition [23].

Input (Di)
Output (Tokens)
Begin
Step 1:
Collect Input documents (Di) where i=1, 2, 3….n;
Step2:
For each input Di;
Extract Word (EWi) = Di;
// apply extract word process for all documents i=1, 2, 3…n in and extract words//
Step 3:
For each EWi;
Stop Word (SWi) =EWi;
// apply Stop word elimination process to remove all stop words like is, am, to, as, etc. //
Stemming (Si) = SWi;
// It create stems of each word, like "use" is the stem of user, using, usage etc. //
Step 4:
For each Si;
Freq_Count (WCi)= Si;
// for the total no. of occurrences of each Stem Si. //
Return (Si);
Step 5:
Tokens (Si) will be passed to an **IR System**.
End

Figure 3. Tokenization Algorithm [21].

The first step of this algorithm is designed to deal with participle and previous plural forms. This step is the most complex and separated into three parts in the original definition. The first part deals with the plural, for example sses→ss and deletion s. The second part deletes ed and ing, or eed→ee if necessary. The second part continues only if ed or ing is deleted and changes the remaining bars to ensure that certain adequacy is recognized later. The third part changes the

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

terminal y i. The remaining steps are relatively easy and contain rules for dealing with various classes of order sufficiency, initially converting double sufficiency into a single suffix and then removing the adequacy of the relevant requirements fulfilled [23].
The fifth step is the transformation of all characters in the document to lowercase letters before the measurement of similarity. We use the cosine similarity function to calculate document similarity [24]. For the two documents di and dj, the similarities between them can be calculated:

$$\cos(d_i, d_j) = \frac{d_i . d_j}{||d_i|| \, ||d_j||} \quad (5)$$

Since the document vectors are of unit length, the above equation is simplified to:

$$\cos(d_i, d_j) = d_i . d_j \quad (6)$$

When the cosine value is 1 the two documents are identical, and 0 if there is nothing in common between them, their document vectors are orthogonal to each other [24]. The sixth step is to transform to lower case for all character in a document.

The final step is clustering. For our analysis, we have chosen the K-means algorithm to group documents. This is a repetitive partitioning process that aims to minimize the least-squares error criteria [25]. As mentioned earlier, Partition clustering algorithms have been recognized to be more suitable for handling large document datasets than hierarchical ones, due to relatively low computational requirements [26-28]. The standard K-means algorithm functions as follows. Given a set of data objects D and the number of k clusters that are predetermined, the data object k is chosen randomly to initialize the cluster k, each being the centroid of a cluster. The remaining objects are then assigned to the cluster represented by the closest or most similar centroid. Next, the new centroid is recalculated for each cluster and in turn, all reassigned documents are based on the new centroid. This step repeats until the convergence solution is still reached, where all data objects remain in the same cluster after the centroid update. The resulting clustering solution is locally optimized for the given data set and initial seed. The choice of different initial seed sets can produce very different end partitions. Methods for finding a good starting point have been proposed [29]. However, we will use the basic K-means algorithm because optimizing of grouping is not main-focus of this paper. The K-means algorithm works by distance steps which basically aim to minimize the distance in the cluster. Therefore, the similarity steps do not directly enter the algorithm, because smaller values indicate differences. K-Main algorithm as follows:

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

1. Select point K as the initial centroid.
2. Set all points to the nearest centroid.
3. Calculate the centroid of each cluster again.
4. Repeat steps 2 and 3 until the centroid doesn't change.

3. RESULTS AND DISCUSSION

We apply the method using RapidMiner Studio Version 7.3.001. Figure 3 shows the model. In this research trial, we used 83 scientific paper document files that we selected randomly, from three categories namely, Deep Learning, Convolutional Neural Network, and Hypertension Retinopathy. The length of the 83 document files varies. Of the 83 document files collected, 12 document files failed to convert to text are id files number 5, 17, 22, 29, 31, 33, 36, 45, 57, 67, 76 and 82.

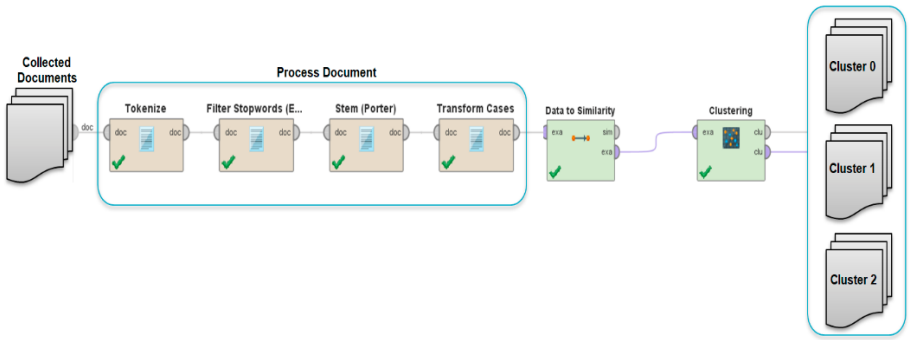

Figure 4. Document Clustering Model

Table 1 shows the results of the clustering process. Of the total 83 document files, 16 categorized files into cluster 0 with the topic Hypertension Retinopathy.

Tabel 1. Clustering Result

Cluster	Topic	Number of Files	File ID	
			True Positive	False Negative
0	Hypertension Retinopathy	16	4,8,12,23,34,49,62,66,73,74,75, 77,79,81, 83	82
1	Convolutional Neural Network	42	1,2,3,6, 10,11,13, 14,16,18, 19,20,21, 25,26,40, 41,42,43, 44,46,47,50, 51,52, 53 56,58,59, 61,65, 70, 71,72,	5,22,31,36,45,48,67, 76
2	Deep	25	7,9,15,24,27,28,30,	17,29,33,57

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Learning		32,35,37, 38,39,54, 55, 60,63,64, 68,69, 78, 80	
Total Number of File	83	70	13

From 16 files as many as 15 files are categorized correctly, while one file is not categorized correctly. While 42 files are categorized into cluster 1 with the topic Convolutional Neural Network (CNN), where 34 files are categorized by correctly and 8 files are not categorized correctly. To measure the accuracy of clustering results, we use the following [24] formula, where accuracy r is defined as:

$$r = \frac{\sum_{i=0}^{2} a_i}{n} \quad (7)$$

where a_i is the number of documents correctly categorized in cluster i and n is the number of documents in the dataset. Based on clustering results in table 1, where 70 documents are correctly classified from a total of 83 documents, the accuracy of clustering is 84.3%. The results of the information retrieval process from 83 documents, consisting of tokenizing, English filter stop-words, Porter stemming and transform all character to lower case, generated 4366 attributes or words with centroid values in cluster 0, cluster 1 and cluster 2. Table 2 shows the results of measurement of the value of centroid in each cluster, for 6 keywords with 2 keywords for each cluster randomly selected. for the words "retinopathi" and "hypertens" the largest centroid values of 0.1033 and 0.1411 are in cluster 0, the results are in accordance with the topic in cluster 0, Hypertension Retinopathy.

Tabel 2. Centroid Value of Each Cluster

Attribute	**Centroid**		
	Cluster 0 (Hypertension Retinopathy)	**Cluster 1** (Convolutional Neural Network)	**Cluster 2** (Deep Learning)
Retinopathi	0.1033	0.0005	0.0000
Hypertens	0.1411	0.0000	0.0000
Cnn	0.0000	0.0433	0.0100
Convolute	0.0012	0.0351	0.0116
Deep	0.0042	0.0281	0.0313
Learn	0.0030	0.0319	0.0337

In the words "cnn" and "convolut" the largest centroid values are 0.0433 and 0.0351 respectively in cluster 1. These results are in accordance with the topic in

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

cluster 1, namely Convolutional neural network or CNN, while for the words "deep" and "learn" the largest centroid value of each 0.0313 and 0.0337 are in cluster 2 the results are in accordance with the topic on the cluster namely deep learning, the value differs slightly from the value in cluster 1, because the word "deep learning" often appear in documents that discuss Convolutional neural networks or CNN.

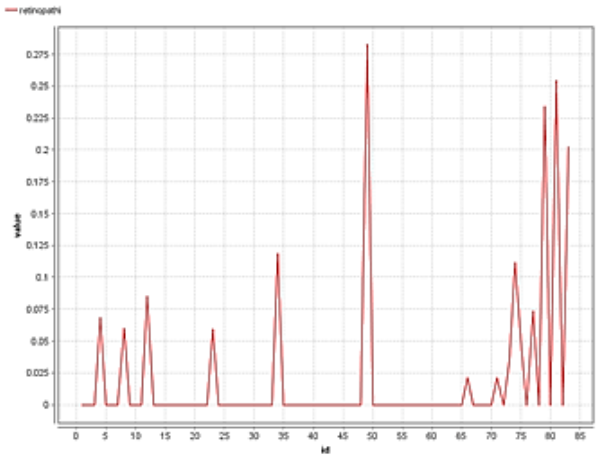

Figure 5. Graph of the index value of the word "retinopathi" in each document

In Figure 5 shows the index value of term frequency and inversed document frequency for the word "retinopathi" where the largest value in the document id is 49, the result is appropriate where the document id 49 is included in cluster 0 with the topic Hypertension Retinopathy.

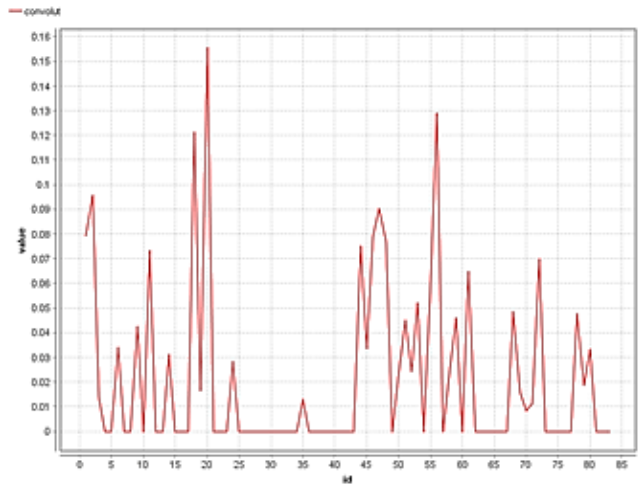

Figure 6. The index value of the word "convolut" in each document.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

While Figure 6 shows the index value of term frequency and inversed document frequency for the word "convolut" where the largest value in the document id 20, the result is also appropriate where document id 20 is included in cluster 1 with the topic Convolutional Neural Network or CNN. Different results are shown in figure 7 where the largest index value for the word "convolut" in document id 52, which is included in cluster 1 with the topic Convolutional Neural Network or CNN, does not cluster 2 with the topic Deep learning. This happens because CNN is one of the deep learning architectural models, so in documents that discuss CNN there are many words of deep learning.

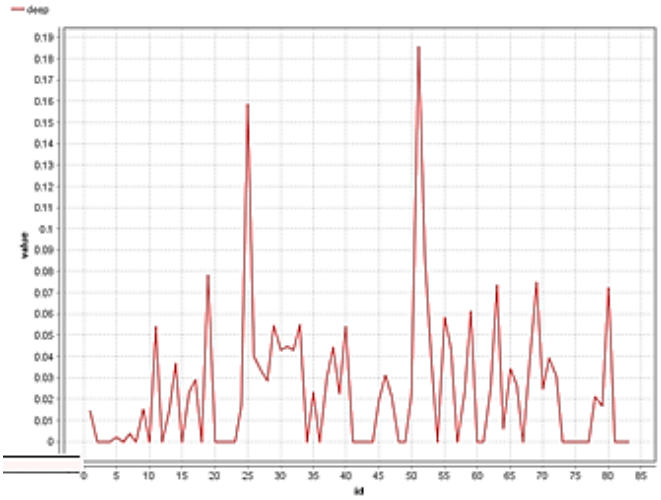

Figure 7. Graph of the index value of the word "deep" in each document.

The index value term frequency and inversed document frequency can be used as an indicator of how many keywords appear in each document so that it becomes information to choose which documents are most relevant to the desired topic.

4. CONCLUSION

This paper presented the results of an experimental study of the grouping technique of 83 scientific document files according to 3 topics are Hypertension Retinopathy, Convolutional Neural Network, and Deep Learning. After the document file has been converted to plain text, then the information retrieval process is tokenized, stop-words English filter, porter stemming and transform all character to lower case. Next, calculate document similarity using cosine similarity. Finally, is the document grouping, we use the K-means standard. Our results show that our method results in an accuracy of 84.3%. The implication of

the results of this study is to get more accurate results using clustering methods using other datasets and document processing methods.

REFERENCES

[1] G. Salton and M. J. McGill, "Introduction to Modern Information Retrieval", McGraw-Hill, 1983.
[2] R. Baeza-Yates, & B. D. A. N Ribeiro, "Modern information retrieval". New York: ACM Press; Harlow, England: Addison-Wesley, 2011.
[3] M.F. Porter. "An algorithm for suffix stripping". *Program*, 14(3), 130-137. 1980.
[4] A.K. Jain, M.N. Murty, & P.J. Flynn. "Data clustering: a review". *ACM computing surveys* (CSUR), 31(3), 264-323. 1999.
[5] P. Willett. "Recent trends in hierarchic document clustering: a critical review". *Information Processing and Management: an International Journal*, 24(5):577–597, 1988.
[6] J. Mcqueen. "Some methods for classification and analysis of multivariate observations". In *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, pages 281–297, Berkeley, CA, 1967.
[7] A. K. Jain and R. C. Dubes. "Algorithms for Clustering Data". Prentice-Hall, Inc., Upper Saddle River, NJ, 1988.
[8] L. Baker and A. McCallum. "Distributional clustering of words for text classification". In *Proc. 1998 Int. Conf. on Research and Development in Information Retrieval (SI- GIR'98)*, pages 96–103, Melbourne, Australia, Aug. 1998.
[9] C. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. "A min-max cut algorithm for graph partitioning and data clustering". In *Proc. 2001 Int. Conf. Data Mining (ICDM'01)*, pages 107–114, San Jose, CA, Nov. 2001.
[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise". In *Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, Portland, Oregon, Aug. 1996.
[11] J. Shi and J. Malik. "Normalized cuts and image segmentation". *IEEE Trans.* on PAMI, 22(8):888–905, 2000.
[12] Andrew Y. Ng, Michael Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In *Advances in Neural Information Processing Systems* 14, pages 849–856. MIT Press, Cambridge, MA, 2001.
[13] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. "Spectral relaxation for k-means clustering". In *Advances in Neural Information Processing Systems* 14, pages 1057–1064. MIT Press, Cambridge, MA, 2001.
[14] Wei Xu, Xin Liu, and Yihong Gong. "Document clustering based on non-negative matrix factorization". In *Proc. 2003 Int. Conf. on Research and*

Development in Information Retrieval (SIGIR'03), pages 267–273, Toronto, Canada, Aug. 2003.

[15] Wei Xu and Yihong Gong. "Document clustering by concept factorization". In *Proc. 2004 Int. Conf. on Research and Development in Information Retrieval (SIGIR'04)*, pages 202–209, Sheffield, UK, July 2004.

[16] A. Anggrawan, K. Hidjah & Q.S. Jihadil. "Kidney failure diagnosis based on case-based reasoning (CBR) method and statistical analysis". In *Informatics and Computing (ICIC), International Conference on (pp. 298-303). IEEE*. 2016.

[17] A. Anggrawan, & A. Azhari. "Aplikasi Deteksi Kemiripan Tugas Paper". *Jurnal Matrik*, 15(2), 5-10. 2016.

[18] G. Salton. "Automatic Text Processing". Addison-Wesley, New York, 1989.

[19] P. K. Chan, D. F. Schlag, and J. Y. Zien. "Spectral k-way ratio-cut partitioning and clustering". *IEEE Trans. Computer-Aided Design*, 13:1088–1096, 1994.

[20] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. harshman. "Indexing by latent semantic analysis". In *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[21] V. Singh and B. Saini, "An effective tokenization algorithm for information", pp. 109–119, 2014.

[22] C. J. van Rijsbergen, "Information Retrieval", 2nd Edition, Butterworths, London, 1979.

[23] M.F. Porter. "An Algorithm for Suffix Stripping", *Program*, 14(3): 130-137. 1980.

[24] N. Sandhya, Y. S. Lalitha, V. Sowmya, K. Anuradha, and A. Govardhan, "Analysis of Stemming Algorithm for Text Clustering", vol. 8, no. 5, pp. 352–359, 2011.

[25] G. Salton. "Automatic Text Processing". Addison-Wesley, New York, 1989.

[26] M. Steinbach, G. Karypis, and V. Kumar. "A Comparison of Document Clustering Techniques". In *KDD Workshop on Text Mining*, 2000.

[27] D. R. Cutting, J. O. Pedersen, D. R. Karger, and J. W. Tukey. "Scatter/gather: A cluster-based approach to browsing large document collections". In *Proceedings of the ACM SIGIR*, 1992.

[28] B. Larsen and C. Aone. "Fast and Effective Text Mining using Linear-time Document Clustering". In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999.

[29] D. Arthur and S. Vassilvitskii. "k-means++ the advantages of careful seeding". In *Symposium on Discrete Algorithms*, 2007.