Study of parameters of the nearest neighbour shared algorithm on clustering documents

Alvida Mustika Rukmi 1, Daryono Budi Utomo2, Neni Imro’atus Sholikhah3
Mathematics Department
Institut Teknologi Sepuluh Nopember (ITS) Indonesia
E-mail: alvidamr@gmail.com1, daryono@matematika.its.ac.id2, nenisholikhah@gmail.com3

Abstract

Abstract. Document clustering is one way of automatically managing documents, extracting of document topics and fastly filtering information. Preprocess of clustering documents processed by textmining consists of: keyword extraction using Rapid Automatic Keyphrase Extraction (RAKE) and making the document as concept vector using Latent Semantic Analysis (LSA). Furthermore, the clustering process is done so that the documents with the similarity of the topic are in the same cluster, based on the preprocessing by textmining performed. Shared Nearest Neighbour (SNN) algorithm is a clustering method based on the number of "nearest neighbors" shared. The parameters in the SNN Algorithm consist of: \( k \) nearest neighbor documents, \( \epsilon \) shared nearest neighbor documents and \( \text{MinT} \) minimum number of similar documents, which can form a cluster. Characteristics The SNN algorithm is based on shared 'neighbor' properties. Each cluster is formed by keywords that are shared by the documents. SNN algorithm allows a cluster can be built more than one keyword, if the value of the frequency of appearing keywords in document is also high. Determination of parameter values on SNN algorithm affects document clustering results. The higher parameter value \( k \), will increase the number of neighbor documents from each document, cause similarity of neighboring documents are lower. The accuracy of each cluster is also low. The higher parameter value \( \epsilon \), caused each document catch only neighbor documents that have a high similarity to build a cluster. It also causes more unclassified documents (noise). The higher the \( \text{MinT} \) parameter value cause the number of clusters will decrease, since the number of similar documents can not form clusters if less than \( \text{MinT} \). Parameter in the SNN Algorithm determine performance of clustering result and the amount of noise (unclustered documents ). The Silhouette coeffisient shows almost the same result in many experiments, above 0.9, which means that SNN algorithm works well with different parameter values.

1. Introduction

Document clustering is to recognize clusters or groups of documents that have the same attribute value [1]. One of the goals of document clustering is to help search information quickly and accurately to document data. The development of information technology, requires a tool to enables representation of the information in accordance with user request.

Text Mining is part of the document clustering, and is the indexing process for database in documents searching. The Text Mining process includes: document keyword extraction and concept extraction. In this paper, the extraction of keyword documents using the Rapid Automatic Keyphrase Extraction (RAKE) method [2]. The RAKE method considers the word association by computing the matrix of blurring with one word against another. The matrix is used to measure the candidate's score and then ranking it [3]. Taking a keyword candidate using an indiviu document-based approach
without relying on document collections, thus speeding up computational time. The process of keyword extraction using the RAKE method has 5 main stages: keyword candidate extraction, calculating the co-occurrence matrix, calculating the value of the ratio, calculating the basic feature value, and selecting the keyword with the highest feature value. Extraction concept using Latent Semantic Analysis method. The process of concept vector formation with LSA method has two stages: construct a weighted matrix of tf-idf and apply Singular Value Decomposition (SVD) method to the matrix formed. The LSA is a method for finding linkages and similarities between documents, fragments and words that appear in documents [4]. Weighted tf-idf performs calculations that describe how important the term (term) in a document. The SVD method is used to minimize the matrix dimension [5]. The result of the SVD decomposition process will give birth to a vector of document concepts.

The clustering algorithm do how to construct clusters of equal size and density, how to cope with noise and how to determine the number of clusters so that the clustering results presenting the members of a cluster bear a high resemblance compared to the different cluster members.

The clustering constraint is generally if the data sets are in large dimensions, the accuracy of the clustering results will decrease, because the similarity between the data objects is higher, making it more difficult for the clustering process [6]. Clustering with the SNN algorithm overcomes it, by determining the nearest residence of all data objects, then the new equality values between the data objects are determined by the number of shared states. The disadvantage of the SNN algorithm is having to specify the tagar parameter value resulting in the desired clustering [7]. In a study conducted by Adriano Moreira, Maribel Y. Santos and Sofia Carneiro stated that the SNN clustering results gave better results for certain datasets than the k-Means algorithm [8]. SNN algorithm requires 3 input parameters as minimum threshold value to build cluster, ie k number of nearest neighbor; ε the number of shared neighbors, and MinT the number of cluster members. After specifying the input parameters, the dokume cluster finds the nearest neighbor of each document. Then the similarity between the two documents is calculated based on the nearest neighbor ε together. By using a similarity measure, the density of each document is calculated as the number of shared neighbors of two documents. Next, the documents are classified as core points, if the density of the document is greater than the core point threshold [9]. Giving optimal parameter values in clustering, capable of handling data in various sizes, controlling cluster quality, and able to handle noise.

Document clustering is to recognize clusters or groups of documents that have the same attribute value [1]. One of the benefits of clustering documents is to help search information quickly and accurately to document data. The development of the world of information, requires a tool capable of presenting the search results information in accordance with user demand. Text Mining is part of the document clustering, and is the indexing process for database searches in the form of documents. The Text Mining process includes: document keyword extraction and concept extraction.

In this paper the extraction of keyword documents using the Rapid Automatic Keyphrase Extraction (RAKE) method [2]. The RAKE method considers the word association by computing the matrix of blurring with one word against another. The matrix is used to measure the candidate's score and then ranking it [3]. Taking a keyword candidate using an indiviudu document based approach is independent of document collections, thereby speeding up computational time. The process of keyword extraction using the RAKE method has 5 main stages: keyword candidate extraction, calculating the co-occurrence matrix, calculating the value of the ratio, calculating the basic feature value, and selecting the keyword with the highest feature value.

While the extraction concept using Latent Semantic Analysis method. The process of concept vector formation with LSA method has two stages: construct a weighted matrix of tf-idf and apply Singular Value Decomposition (SVD) method to the matrix formed. The LSA is a method for finding linkages and similarities between documents, fragments and words that appear in documents.
[4]. Weighted tf-idf performs calculations that describe how important the term (term) in a document. The SVD method is used to minimize the matrix dimension [5].

The clustering algorithm is often confronted with how to construct clusters of equal size and density, how to cope with noise and how to determine the number of clusters so that the clustering results presenting the members of a cluster bear a high resemblance compared to the different cluster members.

The clustering constraint is generally if the data sets are in large dimensions, the accuracy of the clustering results will decrease, because the similarity between the data objects is higher, making it more difficult for the clustering process [6]. Clustering with the SNN algorithm overcomes it, by determining the nearest residence of all data objects, then the new equality values between the data objects are determined by the number of shared states. The disadvantage of the SNN algorithm is having to specify the tagar parameter value resulting in the desired clustering [7].

In a study conducted by Adriano Moreira, Maribel Y. Santos and Sofia Carneiro stated that the SNN clustering results gave better results for certain datasets than the K-Means algorithm [8]. SNN algorithm requires 3 input parameters to construct clusters, i.e. \(k\) number of nearest neighbors, \(\varepsilon\) density threshold (number of shared neighbors), and \(\text{MinT}\) number of core points (cluster members). After specifying the input parameters, the dokume cluster finds the nearest neighbor of each document. Then the similarity between two documents is calculated based on the number of nearest neighbors with it. By using a similarity measure, the density of each document is calculated as the number of shared neighbors of two documents. Next, the documents are classified as core points, if the density of the document is greater than the core point threshold [9]. This study examines the optimal parameter values in clustering that is capable of handling data in various sizes, controlling cluster quality, and able to handle noise.

2. Discussion

2.1 Document keyword extraction using rapid automatic keyphrase extraction (RAKE) method

The experiment uses 525 documents. Clustering the document uses the research title as input on RAKE. The RAKE method has the following stages:

1. Candidate keyword extraction.
   The extraction of a keyword candidate begins by splitting the text using stopword and punctuation. Suppose that is the \(i\) th document of the data. After the document is extracted, we get the keyword candidate \(T_d = \{t_1, t_2, t_3, ..., t_n\}\), where \(t_1, t_2, t_3, ..., t_n\) be words, or phrases of keyword candidates.

2. Build matrix
   After the keyword candidate is obtained, the next step is to calculate the co-occurrence matrix. The co-occurrence matrix represent the frequency of occurrence of a word and keyword phrase.

3. Calculating ratios.
   Ratio value is the ratio between word degree and word frequency. The degree of a word is the number of word occurrences in the document plus the number of phrases that contain the word. The degree of the word on the co-occurrence matrix is obtained from the sum of one column or one line. The word frequency is the number of words in the text. The frequency value can be obtained at the diagonal value of the co-occurrence matrix. The ratio can be calculated using the following Equation 2.1:
   \[
   \text{ratio} = \frac{\text{deg}(w)}{\text{freq}(w)} \tag{2.1}
   \]

   where :
   \[\text{deg}(w) = \text{degree of word}\]
   \[\text{freq}(w) = \text{frequency of word}\]

4. Calculating the value of basic features.
   The basic feature value is the sum of the existing keyword ratios in the keyword candidate. The value is then sorted by the highest value to the lowest. The basic feature values are calculated using the following Equation 2.2
   \[Nfd = \text{ratio} (t) + \text{ratio} (f) \tag{2.2}\]
where: 
\[ Nfd = \text{value of basic features} \]
\[ \text{ratio (t)} = \text{keyword ratio of t} \]
\[ \text{ratio (ft)} = \text{phrase ratio containing the keyword of t} \]

2.2 Concept extraction using latent semantic analysis (LSA) method

The steps of concept vector formation are as follows:

\[ W_{ij} = t_{ij} \times idf_j = t_{ij} \times \log \left( \frac{N}{df_j} \right) \]  
(2.3)
\[ idf = \log \frac{N}{df} \]  
(2.4)

Figure 1. Building vector process.

Term Weighting aims to determine the weight of each term. The calculation of term weight requires two things: Term Frequency (tf) and Inverse Document Frequency (idf). Term Frequency (tf) is the frequency of occurrence of a term (term) in a document. The value of tf varies across each document depending on the appearance of a word in a document. The value of tf is proportional to the rate of occurrence of the term in the document. The more often the term appears on a document, the greater the value of tf in the document and the less the term appears the smaller the value of tf. In addition to Term Frequency is also required Inverse Document Frequency (idf) on the weighting term. Inverse Document Frequency (idf) is the frequency of occurrence of term-i in the whole document. The Equation 2.3 is calculate the value of tf-idf:

\[ W_{ij} = t_{ij} \times idf_j = t_{ij} \times \log \left( \frac{N}{df_j} \right) \]  
(2.3)
\[ idf = \log \frac{N}{df} \]  
(2.4)

where:
\[ D_1, \ldots, D_n = \text{document} \]
\[ W_{ij} = \text{weight } i^{th}-\text{term to } j^{th}-\text{document} \]
\[ t_{ij} = \text{number of occurrences of } i^{th}-\text{term into } j^{th}-\text{document} \]
\[ N = \text{total number of documents} \]
\[ df_j = \text{number of documents containing term-i} \]
\[ idf_j = \text{Inverse document frequency contain term-i} \]

SVD will decompose a matrix into three new matrices of left singular vector matrix, singular value matrix and right singular vector. The SVD of an mxn dimension, then matrix A is as follows

\[ A_{mxn} = U_{m \times n} S_{n \times n} V_{n \times d}^T \]  
(2.5)

where:
\[ A = \text{Matrix by mxd dimension} \]
U = left singular vector matrix by mxn dimension  
S = singular value matrix by nxn dimension  
V = right singular vector dxn dimension  

2.3 Clustering by shared nearest neighbor (SNN) algorithm

In this process the extracted concepts of the document are grouped into clusters representing a topic. This clustering uses the SNN method. At this stage insert the input parameter of \( k \) list of neighbor, \( \varepsilon \) of radius of neighbor region \( N_\varepsilon (p) \) from a data point \( p \) and \( MinT \) minimum number of interior points in region \( N_\varepsilon (p) \). The following is a drawing of the document cluster flow diagram with the SNN algorithm:

![Diagram flow of document clustering by SNN algorithm.](image)

Then, carry out of the result preprocess text mining using by RAKE and LSA and then simulate using parameter in SNN Algorithm to cluster documents. So, characteristic of parameter in SNN algorithm will be study. The experiment were described below:
2.3.1 Determine $K = 3$

Table 1. Document clustering by $k=3$.

| $\varepsilon$ | MinT | Total Cluster | Doc Clustered | Doc Noise | Percentage Noise/Total doc | Coefficient Silhoute |
|---------------|------|---------------|---------------|-----------|---------------------------|---------------------|
| 2             | 2    | 116           | 413           | 112       | 21.3%                     | 0.987840123         |
| 2             | 3    | 63            | 307           | 218       | 41.5%                     | 0.98666718495585    |
| 2             | 4    | 41            | 241           | 284       | 45.9%                     | 0.986599355602242   |
| 3             | 2    | 35            | 79            | 446       | 84.9%                     | 0.9820522612131688  |
| 3             | 3    | 9             | 27            | 498       | 94.8%                     | 0.9808768419862584  |

2.3.2 Determine $K = 5$

Table 2. Document clustering by $k=5$.

| $\varepsilon$ | MinT | Total Cluster | Doc Clustered | Doc Noise | Percentage Noise/Total doc | Coefficient Silhoute |
|---------------|------|---------------|---------------|-----------|---------------------------|---------------------|
| 2             | 2    | 10            | 514           | 11        | 2.09%                     | 0.98926368          |
| 2             | 3    | 3             | 500           | 25        | 4.76%                     | 0.988637763         |
| 2             | 4    | 3             | 500           | 25        | 4.76%                     | 0.988637763         |
| 3             | 2    | 82            | 342           | 183       | 34.8%                     | 0.987840123         |
| 3             | 3    | 45            | 268           | 257       | 48.9%                     | 0.986534609269849   |
| 3             | 4    | 9             | 220           | 305       | 58.1%                     | 0.986074886         |

2.3.3 Determine $K = 10$

Table 3. Document clustering by $k=10$.

| $\varepsilon$ | MinT | Total Cluster | Doc Clustered | Doc Noise | Percentage Noise/Total doc | Coefficient Silhoute |
|---------------|------|---------------|---------------|-----------|---------------------------|---------------------|
| 2             | 2    | 1             | 524           | 1         | 0.19%                     | 0.991300821         |
| 2             | 4    | 1             | 524           | 1         | 0.19%                     | 0.991300821         |
| 2             | 6    | 1             | 524           | 1         | 0.19%                     | 0.991300821         |
| 7             | 2    | 47            | 174           | 351       | 66.8%                     | 0.985487083         |
| 7             | 4    | 14            | 99            | 426       | 81.1%                     | 0.985088194         |
| 7             | 6    | 7             | 70            | 456       | 86.8%                     | 0.9810475153652917  |

2.3.4 Determine $K = 15$

Table 4. Document clustering by $k=15$

| $\varepsilon$ | MinT | Total Cluster | Doc Clustered | Doc Noise | Percentage Noise/Total doc | Coefficient Silhoute |
|---------------|------|---------------|---------------|-----------|---------------------------|---------------------|
| 3             | 2    | 1             | 523           | 2         | 0.38%                     | 0.990178031         |
| 3             | 4    | 1             | 523           | 2         | 0.38%                     | 0.990178031         |
| 3             | 6    | 1             | 523           | 2         | 0.38%                     | 0.990178031         |
| 10            | 2    | 46            | 186           | 339       | 64.5%                     | 0.986044764         |
| 10            | 4    | 14            | 116           | 409       | 77.9%                     | 0.985424659         |
| 10            | 7    | 8             | 88            | 437       | 83.2%                     | 0.984802256         |
| 12            | 2    | 21            | 82            | 443       | 84.3%                     | 0.984374682802898   |
| 12            | 6    | 3             | 34            | 491       | 93.5%                     | 0.98130261101419    |
| 12            | 12   | 1             | 15            | 510       | 97.1%                     | 0.9804164843922515  |

The value of $k$ must be less than or equal to $\varepsilon$ value. Based on the experimental results, the parameter value of $k$ on documents clustering does not always result in a large number of cluster, depending on the $\varepsilon$ and $\text{MinT}$ values.

Determine $k$ value above 10, if the value of $\varepsilon$ is small, tends to produce a single cluster, even any $\text{MinT}$ value has no effect. While the determination of the small value of $k$ (less than 1% of the amount
of data) tends to represent more large number of clustered documents than value of $k > 1\%$ of the amount of data. When $\varepsilon$ approach to $k$ value, it will increase number of clusters according by decreasing of $\text{MinT}$ value. Giving a large value of $\varepsilon$, will form clusters that contain documents with high topic similarities. This results in each cluster having a high accuracy value because the relevance of the document to the keyword is also high, so that only a few irrelevant documents appear on each cluster indicate a minimum threshold of the number of documents in a cluster. Based on the experiment show that greater $\text{MinT}$ value will form t less the number of clusters.

The noise document is an unclassified document. The percentage noise shows the percentage ratio of unclassified documents against the total number of documents. This percentage increases when the difference in $k$ value with $\varepsilon$ value reduce, moreover $\text{MinT}$ increases. In the experiments show that smaller $\varepsilon$ will give the result smaller percentage of noise. Small noise percentage value means to show more documents that can be clustered.

Silhouette coefficient refers to a method of interpretation and validation of consistency within clusters of data. The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from $-1$ to $+1$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters. The Silhouette coefficient shows almost the same result in many experiments, above 0.9, which means that SNN algorithm works well with different parameter values.

3. Conclusion

Here are the conclusions obtained during the clustering process with the SNN algorithm:

1. Each cluster is formed by keywords that are shared by documents in a cluster. SNN algorithm allows a cluster can be built more than one keyword, if the value of the frequency of along with keywords is also high. Thus, a cluster is formed based on the keywords used together by the documents in there.

2. Determination of parameter values on the SNN algorithm will affect the results of document clustering, in the form of:
   a. The value of $k$ must be less than or equal to $\varepsilon$ value so it form clustering. The higher $k$ value will increase the number of nearest neighbor documents for each document. It can decrease similarity of neighboring documents. The clustering results show that the low similarity values between documents will be in one cluster.
   b. High $\varepsilon$ value cause more noise, but among documents in one cluster show a high similarity. The high $\varepsilon$ value results in the document only 'pulling' neighboring documents have a high similarity when forming a cluster.
   c. The $\text{MinT}$ value limits the number of clusters based on the minimal number of documents that can form a cluster. If the $\text{MinT}$ value is high, the number of clusters will decrease, since the number of documents less than MinT can not form a cluster.

Determination value of this parameter depends on requirement of a problem. In the document clustering requires a low $k$ value, a high $\varepsilon$ value ($k > \varepsilon$) and low $\text{MinT}$ value. It can represent a good clustering.

References

[1] Peng, Y., Kou, G., Chen, Z., Shi, Y., 2006, Recent Trends in Data Mining (DM) :Document Clustering of DM Publications, *Int'l Conference on Service Systems and Service Management*, vol. 2, pp.

[2] Berry, M.W., dan Kogan, J., 2010, Text Mining : Applications and Theory. *John Willey & Sons*.

[3] Lee, D.L., 1997, Document Ranking and the Vector-Space Model. *Hong Kong University of Science and Technology*.

[4] Luthfiarta, A., Zeniarja, J., Salam, A., 2013, Integrasi Peringkas Dokumen Otomatis Dengan
Latent Semantic Analysis (LSA) pada Peringkas Dokumen Otomatis untuk Proses Clustering Dokumen, *Seminar nasional teknologi informasi & komunikasi terapan*.

[5] Salton G., dan Buckley C., 1998, Term Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management*, vol.24, no. 5, hal :513-523.

[6] Heidelberg, 2005, High Dimensional Shared Nearest Neighbor Clustering Algorithm”, Lecture Notes on Computer Science, Publisher Springer Berlin vol.3614, Hal. 494-50

[7] Ertöz L., Steinbach M., dan Kumar V.2002, Finding topics in document, a Shared Nearest Neighbor Approach Clustering and Information Retrieval, *Kluwer Academic Publisher*.

[8] Moreira Adriano, Santos Maribel Y and Carneiro Sofia, 2005, Density-based clustering algorithms – DBSCAN and SNN, *University of Minho – Portugal*.

[9] Jarvis R.A. dan Patrick E. 1973, Clustering Using a Similarity Measure based on Shared Nearest Neighbor, *Proceeding IEEE Transaction on Computer*, vol C-22, hal. 1025-1034.

[10] Rukmi, A.M., Irawan M.I, Hidayat, N., 2009, Pengklasteran Data Kategoris dengan Algoritma Shared Nearest Neighbor, *Institut Teknologi Sepuluh Nopember*, Vol. 6, No. 1