Asian Journal of Information Technology 10 (8-12): 341-347, 2011
ISSN: 1682-3915

The Use of Hartigan Index for Initializing K-Means++ in Detecting Similar Texts of Clustered Documents as a Plagiarism Indicator

Diana Purwitasari, I. Wayan Surya Priantara, Putu Yuwono Kusmawan,
Umi Laili Yuhana and Daniel Oranova Siahaan
Department of Informatics Engineering, Faculty of Information Technology,
Institut of Teknologi Sepuluh Nopember (ITS) Jl. Raya ITS,
Ged. T. Informatika ITS Sukolilo, 60111 Surabaya, Indonesia

Abstract: Plagiarism is increasingly alarming, especially if this happens in the field of education. Many writing works in which a part of the content is written by plagiarizing other people's works. Similar sentence detection as a plagiarism indicator can be conducted by using n-gram based hashing algorithm of Winnowing algorithm. The function of Winnowing is to generate document fingerprint which convert texts within document into a collection of hash values. Similar fingerprint between documents shows that there are similar texts as a plagiarism indicator. Plagiarizing usually happens on documents having similar topics. Therefore, to detect plagiarism, documents having similar topics should be clustered. K-means++ is a clustering algorithm that requires cluster number as its input through recommendation conducted by Hartigan index to give a recommendation for the cluster number. After clustering documents, a comparison was made between document fingerprint and fingerprint cluster instead of between documents. Then, the comparison was made for documents which become members of the closest cluster that had been selected from the first comparison.

Key words: Plagiarism detection, document fingerprint, winnowing, K-means++, hartigan index, indicator

INTRODUCTION

Internet has made a lot of online information easily available. The need for searching references can be easily carried out and this has resulted in the increasing number of misused copy paste activities leading to plagiarism. In the field of education, plagiarism has often been conducted by students working on scientific writing. From lecturer's perspective, copy paste activities have made evaluation difficult although, there is already a clear statement about the sanction of plagiarism. Plagiarism activities will be more difficult to be detected if there is a lot of documents that need to be reviewed. For that reason, the works of plagiarism detection in this study are required to help lectures to evaluate students' tasks or assignments. Similar text detection as a plagiarism indicator was conducted to find out the percentage level of similarities among documents. Thus, it can be found out whether someone has carried out plagiarism. The activity of detecting similar text is known as common subsequence problem (Oetsch *et al*., 2010).

A scientific writing can be said to conduct plagiarism if taking other people's work without attributing the work to the original writer. It was assumed that plagiarism activity was categorized into two types: global and local. Global plagiarism happens if a part of writing works comes from the sources outside the school or university which are used without acknowledging the original writer. The sources of global plagiarism are internet, books and etc. Plagiarism detection methods have been developed from very simple single-user desktop application to internet based one. Meanwhile, plagiarism can be said to have local type if a part of the same writing works came from the sources from the same school/or university. The sources of local plagiarism can be from students' assignments of the same class or those of previous year.

Local type of plagiarism is common in developing countries where internet access is limited or expensive. The most well-known internet based application for plagiarism detection was Turnitin (Butakov and Scherbinin, 2009). Educational organizations or institutions can subscribe to get direct access or use Turnitin services using plug-in. Turnitin compares documents inserted by users with the ones already available on the database and it will give a report showing their level of similarities. Another global type of plagiarism

Corresponding Author: Diana Purwitasari, Department of Informatics Engineering, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember (ITS) Jl. Raya ITS, Ged. T. Informatika ITS Sukolilo, 60111 Surabaya, Indonesia

detection toolboxes are Dustball and Copyscape. Dustball uses searching machine facilities to find texts or sentences that is assumed to become the result of web plagiarism while copyscape detects the content of web page according to the inserted URL address. Both applications assume plagiarism on the basis of word order position in sentences or texts. Applications which can be able to detect local plagiarism among others are YAP3 (Wise, 1996) and CopyCatch.

This research discussed about another approach of detecting similar texts from large collection of documents in order to find out whether students have carried out local plagiarism. Winnowing algorithm was used to search the level of text similarities between documents as an indicator of plagiarism (Schleimer *et al.*, 2003). Winnowing algorithm is an algorithm which converts string texts into hash values called fingerprint. The fingerprint values of each document are used to identify text similarities between documents. They can also be used to cluster documents (Parapar and Barreiro, 2009). In this study, we used K-means++ which is an improvement of K-means algorithm, to perform the clustering and Hartigan index to give a recommendation for the cluster number. Evaluation of similar texts detection as a plagiarism indicator was performed by checking the works of students of informatics engineering department at 2009-2010 who were studying socio ethics. The concerns were about accuracy and processing time for plagiarism detection.

WINNOWING ALGORITHM FOR CREATING DOCUMENT FINGERPRINT

There are many methods used to detect plagiarism in documents by recognizing similar sentences. However, some basic requirements must be fulfilled by the algorithm, namely (Schleimer *et al.*, 2003), whitespace intensity which means that the identification for similar sentences should not be influenced by space, font (capital or normal), punctuations etc., noise suppression which means that the identification for similar sentences is not influenced by word length such as an article, position independence which means that the similarity finding should not depend on position of words thus, word order in different position can still be recognized ada ide yg terputus Manber bitap algorithm (Muth and Manber, 1996) and Winnowing algorithm (Schleimer *et al.*, 2003) use n-gram based hashing algorithm for string matching that consider the issues of whitespace intensity, noise suppression and position independence. The algorithms can tell whether a given text contains a substring (the same as common subsequence problem) which is approximately equal to a given pattern or fingerprint. The input is the processed text document and the output is a collection of hash values called fingerprint. The hash values are numeric values created from the calculation of ASCII of each character in the texts. The difference between Manber and Winnowing algorithms lie on the selection of hash values as document fingerprint. Winnowing algorithm uses window while Manber algorithm uses modulus. However, Winnowing algorithm is more informative than Marber algorithm in that it can save position of hash values. The first step of Winnowing algorithm application is to discard the characters of the:

The classic problem in machine learning
↓
The classicic problem in machine earning

irrelevant document content such as punctuation, space and other symbols. For example; the next step is forming gram value series from the cleaned document when N = 5:

The classic problem in machine learning
↓

thecl	hecla	eclas	class
lassi	assic	ssicp	sicpr
icpro	cprob	probl	roble
oblem	blemi	lemin	eminm
minma	inmac	nmach	machi
achin	chine	hinel	inele
nelea	elear	learn	earni
arnin	rning		

The third step is forming hash values of ACII values of each character from the created gram series using rolling hash equation. This step uses Eq. 1:

$$H_{(c_1 \dots c_k)} = c_1 \times b^{(k-1)} + c_2 \times b^{(k-2)} + \dots + c_{(k-1)} \times b^{k+c_k} \quad (1)$$

Where:

c = ASCII values of the characters
b = Prime number basis
k = Number of characters

An example of the creation of hash value of n-gram thecl using Eq. 1 with basis value of 3 is as follows:

$$H_{(\text{thecl})} = \text{ascii}(t) \times 3^{(4)} + \text{ascii}(h) \times 3^{(3)} + \text{ascii}(e) \times 3^{(2)} + \text{ascii}(c) \times 3^{(1)} + \text{ascii}(l) \times 3^{(0)}$$

$$H_{(\text{thecl})} = 116 \times 81 + 104 \times 27 + 101 \times 9 + 99 \times 3 + 105 \times 1 = 13518$$

The advantage of using rolling hash is that the following hash $H_{(c_2 \ldots c_{k+1})}$ value can be obtained using Eq. 2:

$$H_{(c_2 \ldots c_{k+1})} = (H_{(c_1 \ldots c_k)} - c_1 \times b^{(k-1)} \times b + c_{(k+1)} \tag{2}$$

An example of hash value creation utilizing the formed hash value using n-gram hecla is:

$$\begin{aligned} H_{(\text{hecla})} &= (13518 - \text{ascii}(t) \times 3^{(4)} \times 3 + \text{ascii}(a) \times 3^{(0)} \\ &= (13518 - 116 \times 81) \times 3 + 97 \times 1 \\ &= 12463 \end{aligned}$$

Finally, the created hash values are made windows then the smallest hash value of each window is selected to be the fingerprint of each document:

thecl	hecla	eclas	class
lassi	assic	ssicp	sicpr
		↓	
13518	12463	12232	12268
12852	12411	13774	13491

Ex: for windows with size w = 4, [thecl hecla eclas class] will make [13518 12463 12232 12268]. Complete windows of hash values for document text the classic problem in machine learning are:

[13518 12463 **12232** 12268] [12463 **12232** 12268 12852]
[**12232** 12268 12852 12411] [**12268** 12852 12411 13774]
[12852 **12411** 13774 13491] [**12411** 13774 13491 12639]
[13774 13491 12639 **12500**] [13491 12639 **12500** 13551]
[12639 **12500** 13551 13538] [**12500** 13551 13538 13021]
[13551 13538 13021 **12195**] [13538 13021 **12195** 12881]
[13021 **12195** 12881 12508] [**12195** 12881 12508 13078]
[12881 **12508** 13078 12846] [**12508** 13078 12846 13127]
[13078 12846 13127 **12756**] [12846 13127 **12756** 11891]
[13127 **12756** 11891 12203] [**12756** 11891 12203 12660]
[**11891** 12203 12660 12809] [**12203** 12660 12809 13009]
[12660 12809 13009 **12411**] [12809 13009 **12411** 12800]
[13009 **12411** 12800 12261] [**12411** 12800 12261 12350]
[12800 **12261** 12350 13582]

So, the produced fingerprint values are:

12232 12268 12411 12500 12195 12508 12756
11891 12203 12411 12261

These values were then used to find the percentage level of similarities of one document to another using Jaccard Coefficient equation which is shown by Eq. 3:

$$\text{similarity}(d_i, d_j) = \frac{|W(d_i) \bigcap (W d_j)|}{|W(d_i) \bigcup (W d_j)|} \tag{3}$$

W (d_i) and W (d_j) represent fingerprint values of document i and j. W (d_i), ∪W (d_j) are the combination of fingerprint values of both documents. Meanwhile, W (d_i) ∩W (d_j) constitutes the pieces of fingerprint values of both documents. For example:

Fingerprint D$_1$ =
11891 12203 12411 12261 12350 12803 12351 12135 12211 12450 13351 12377 12891 12114 12497
Fingerprint D$_2$ =
12232 12268 12411 12500 12195 12508 12756 11891 12203 12411 12261

$$\begin{aligned} \text{Similarity}(d_i, d_j) &= \frac{|11891\ 12203\ 12411\ 12261|}{21} \\ &= \frac{4}{21} = 0.1905 \end{aligned}$$

Winnowing algorithm requires basis value, window size and n-gram value. The three parameters were then experimented to find the best combination to detect plagiarism. In addition to the three parameters, the algorithm needs tolerance threshold value of plagiarism.

USAGE OF HARTIGAN INDEX AND K-MEANS++ FOR FINGERPRINT-BASED DOCUMENTS CLUSTERING

Clustering is a method of analyzing data aiming to group or cluster data having the same characteristics into one group or cluster and the one having different characteristics into another group. There are several approaches used to develop clustering. Two of them are partition and hierarchical-based approaches. Partition based approach also called partition-based clustering is grouping data by selecting the analyzed data to the available clusters. While hierarchical-based approach also called hierarchical-based clustering is grouping data by making a hierarchy in the form of dendogram where the similar data is put on the closer hierarchy and not on the farther one.

K-Means clustering algorithm: These are steps of clustering using K-Means algorithm with some adjustments for calculating distance between document items consisting of hash values (Parapar and Barreiro, 2009).

Step 1: Determine k value of the clusters which will be created or set.

Step 2: Determine an initial centroid randomly as many as the number of clusters which will be set. Centroid is a collection of hash values. Initially, the centroid values will be the same as fingerprint values of the selected document.

Step 3: Calculate the similarity level of each document of each centroid using Eq. 3.

Step 4: Group or cluster every document to the closest cluster so that documents having similar topics lie in the same cluster and those having different topics in different cluster.

Step 5: Determine hash values as the new centroid using Eq. 4:

$$\text{Centroid}(C) = \bigcup f(C,h) \,|\, h \in \bigcup W(d_i)_{i=1...n}; \frac{hf(h,C)}{n} \geq \gamma \tag{4}$$

Centroid C contains a collection of hash h. Assuming that cluster C has document members $d_i...d_n$, thus statement of $h \in \bigcup W(d_i)_{i=1..n}$ which explained about temporary selected h were derived from W $(d_i) \cup ... \cup W$ (d_n). The selected hash h should meet the terms with function f (C, h), so that where h f (h, C)/n$\geq\gamma$ where h f (h, C) is the occurrences number of hash h within all documents in one cluster C. Threshold value for γ is set from the user's input. In short f (C, h) was a function for a hash h which was observed in cluster C. The equation statement $\cup$f (C, h) means that all of hash h which complies f (C, h) will be combined and become centroid C.

Step 6: Go back to step 3 if the hash values in the new centroid and the old one are not the same. This means clustering stops when old and new centroid values remain unchanged.

K-Means++ clustering algorithm: K-means++ is developed from K-Means algorithm which is one of clustering methods using partition-based approach. K-Means selects initial values or seeds of centroids randomly which sometimes result in a longer processing time. Therefore, K-means++ is used to reduce longer-time process weakness of K-Means (Arthur *et al.*, 2007). The following are the steps of clustering using K-Means++ algorithm.

Step 1: Determine one initial centroid of the whole document data randomly using uniform distribution. Let assume the centroid will become a document called d_c. For example there are three documents, d_1-d_3. Here the randomly selected centroid d_{c1} will be d_2.

Step 2: For every document file d_i, calculate dissimilarities between document file and the closest centroid selected using Eq. 5. The dissimilarity value is obtained from substraction of 1 and similarity value from Eq. 3.

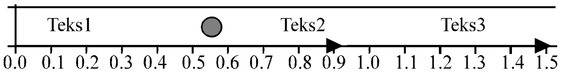


Fig. 1: The value of rages Teks1-3

$$\text{dissimilarity}(d_i, d_c) = D(d_i, d_c) = 1 - \text{similarity}(d_i, d_j)$$
$$= 1 - \frac{|W(d_i) \bigcap W(d_j)|}{|W(d_i) \bigcup W(d_j)|} \tag{5}$$

Step 3: Add new centroid from the unclustered documents using weighted probability distribution of $D(d_c^2)$ in Eq. 6.

$$D(d_c^2) = \sum_{i=1...n} D(d_i, d_c)^2 \tag{6}$$

Values of dissimilarities between documents and the first centroid will be:

$$D\ (d_1, d_{c1})^2 = 0.924;\ D\ (d_3, d_{c1})^2 = 0.605$$

And:

$$D(d_{c1}^2) = 1.529$$

Therefore, range value between 0.000-0.924 belongs to document d_1 sor Teks1. There is no range value for document d_2 or Teks2 and range value for d_3 or Teks3 is 0.924−(0.924+0.605) equals to 0.924-1.529.

For second seed of next centroid, randomly select value between 0.000-1.529 and the result is 0.563. The value of 0.563 is within range of Teks1 so that d_{c2} will be d_1 as shown in Fig. 1.

Step 4: Repeat step 2 and 3 until some k centroids have been selected.

Step 5: Do clustering using K-Means algorithm with these centroids as initial seeds.

Hartigan index: Grouping or clustering requires a number of groups which will be created or set and this is derived from the users' inputs. However, the users' inputs might have underset value or overset value. Therefore, algorithms such as Rule of Thumb and Hartigan Index are required to determine the number of clusters or groups which should be set from the available data items. Rule of Thumb in Eq. 7 is used to determine the number of clusters based on n number of data items:

$$k \approx \sqrt{n/2} \tag{7}$$

Where:

k = The number of clusters
n = The number of data items which will be clustered

For example, when the number of data is 100, the k value using Eq. 7 is 7.071, so the recommended number of clusters is eight.

Hartigan Index in Eq. 8 as a statistical method functions to select the approriate value of k as a result it can produce the best means across clusters:

$$H(k) = (n - k - 1)\frac{err(k) - err(k+1)}{err(k+1)} \quad (8)$$

Where:

k = The number of clusters
n = The number of data items

Meanwhile, err value in Eq. 9 is an accumulative value of document dissimilarity level to the closest centroid. Initially, use k value from Eq. 7 (Rule of Thumb) to set the recommended cluster number. Then, analyze the appropriateness of k value using Hartigan Index. Assuming that the k value range is k±3. Thus, the cluster number which is supposed to be set is k with maximum value of H (k) from Eq. 8.

$$err(k) = \sum_{i=1}^{k} \sum_{j=1;\, j\varepsilon Ci}^{n} D(d_j, d_{ci})^2 \quad (9)$$

EXPERIMENT AND EVALUATION

Researchers implemented aforementioned steps to evaluate detection system of similar texts as a plagiarism indicator. The architecture of the system is shown in Fig. 2. K-Means++ algorithm was used in partition clustering while Winnowing algorithm helped search the similarities between one document to another. Detection began with document reading to extract file content which would be checked and converted into strings. Here, plagiarism detection was focused on document file with extension doc, docx and pdf.

Meanwhile, Apache PDFBox Library (in JAVA programming language, http://pdfbox.apache.org/) was used for reading.pdf whilst Apache POI Library (Java API for Microsoft Documents, http://poi.apache.org/) for .doc and .docx.

Menu [List Document] in Fig. 3 shows existing documents in the database and authenticity level of document contents obtained with Eq. 10. Authentic value of 99.67% means that the observed document only had 0.33% similar sentences which were probably resulted from copy-paste activities.

Users can change cluster names in Menu [List Clusters] based on their keywords shown in the Menu [Keywords]. Cluster 0 generally had documents items that could not be grouped into any existing clusters (#1 ... #3). Menu [Settings] was used to do configuration value setting. Variation of configuration values could give better or worse results of similar texts detecting:

$$authenticity(d_i, d_j) = \frac{\#different\ hash\ number}{\#total\ hash\ number} \times 100\% \quad (10)$$

Configuration values for experiments on plagiarism detection were:

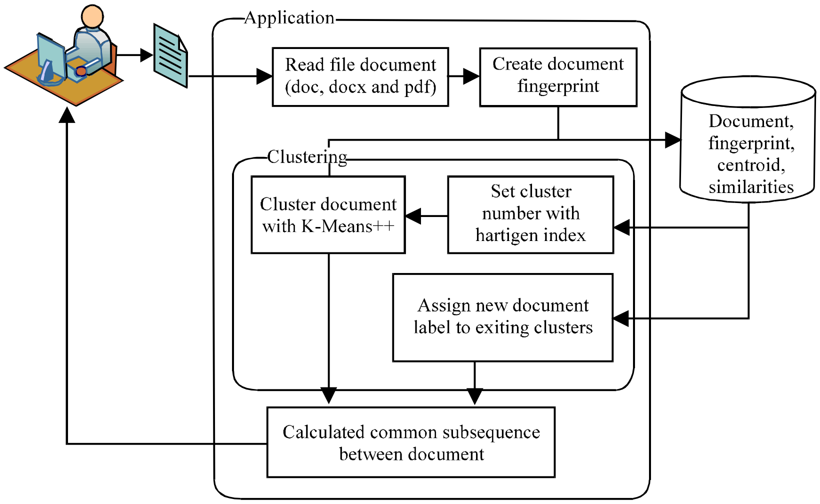
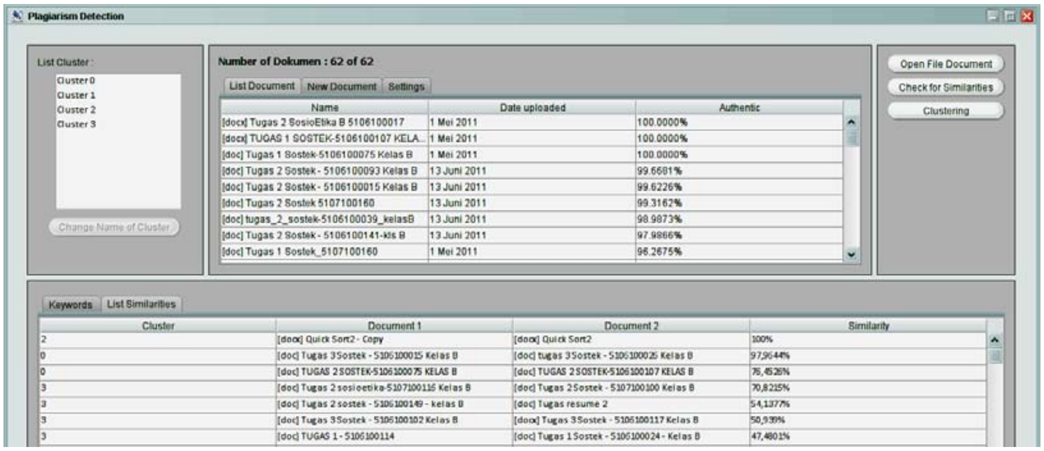

Application
Read file document (doc, docx and pdf)
Create document fingerprint
Document, fingerprint, centroid, similarities
Clustering
Cluster document with K-Means++
Set cluster number with hartigen index
Assign new document label to exiting clusters
Calculated common subsequence between document

Fig. 2: Architecture of detecting similar texts as a plagiarism indicator



Number of Dokumen : 62 of 62

Name	Date uploaded	Authentic
[docx] Tugas 2 SosioEtika B 5106100017	1 Mei 2011	100.0000%
[docx] TUGAS 1 SOSTEK-5106100107 KELA...	1 Mei 2011	100.0000%
[doc] Tugas 1 Sostek-5106100075 Kelas B	1 Mei 2011	100.0000%
[doc] Tugas 2 Sostek - 5106100093 Kelas B	13 Juni 2011	99.6681%
[doc] Tugas 2 Sostek - 5106100015 Kelas B	13 Juni 2011	99.6226%
[doc] Tugas 2 Sostek 5107100160	13 Juni 2011	99.3162%
[doc] tugas_2_sostek-5106100039_kelasB	13 Juni 2011	98.9873%
[doc] Tugas 2 Sostek - 5106100141-kls B	13 Juni 2011	97.9866%
[doc] Tugas 1 Sostek_5107100160	1 Mei 2011	96.2675%

Cluster	Document 1	Document 2	Similarity
2	[docx] Quick Sort2 - Copy	[docx] Quick Sort2	100%
0	[doc] Tugas 3 Sostek - 5106100015 Kelas B	[doc] tugas 3 Sostek - 5106100025 Kelas B	97,9644%
0	[doc] TUGAS 2 SOSTEK-5106100075 KELAS B	[doc] TUGAS 2 SOSTEK-5106100107 KELAS B	76,4526%
3	[doc] Tugas 2 sosioetika-5107100116 Kelas B	[doc] Tugas 2 Sostek - 5107100100 Kelas B	70,8215%
3	[doc] Tugas 2 sostek - 5106100149 - kelas B	[doc] Tugas resume 2	54,1377%
3	[doc] Tugas 3 Sostek - 5106100102 Kelas B	[docx] Tugas 3 Sostek - 5106100117 Kelas B	50,939%
3	[doc] TUGAS 1 - 5106100114	[doc] Tugas 1 Sostek - 5106100024 - Kelas B	47,4801%

Fig. 3: User interface for detecting similar texts as a plagiarism indicator

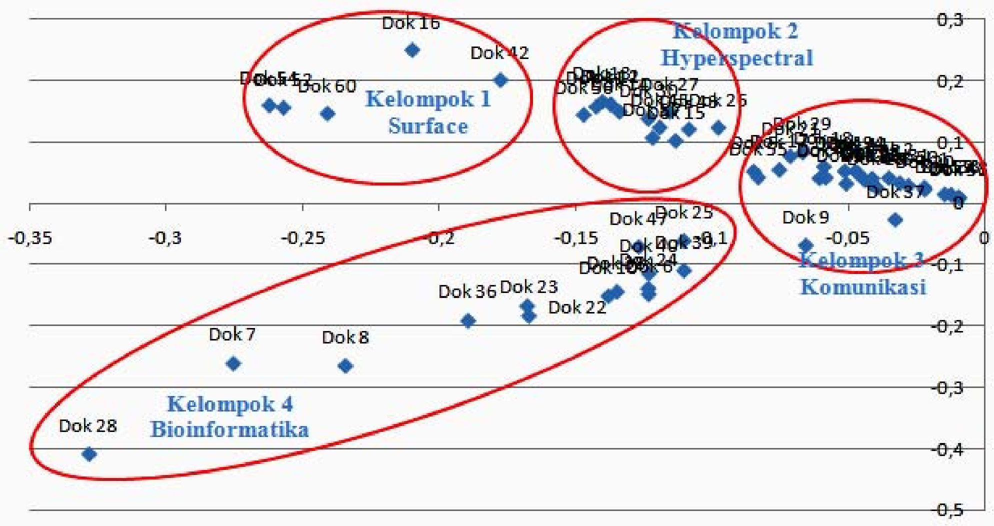


Fig. 4: Result of analysing topics in experiments data with LSA

- n value for n g
- Prime number b as a basis of hashing function
- Window size w to identify the more appropriate fingerprint
- Threshold value to limit tolerance of similar sentences

The previous research showed that values of n = 30, b = 3, w = 30 and threshold = 50% could give better results for detecting similar texts as a plagiarism indicator. The experiment data contained 60 documents of course reports made by students of informatics engineering department at 2009-2010 who were studying socio ethics. The reports was divided into four topics: communication, bioinformatics, surface and hyperspectral. Analysis of the existence of those four topics was carried out by Latent Semantic Analysis (LSA) with Singular Value Decomposition (SVD) technique (Landauer *et al*., 1998). Figure 4 shows 2-rank approximation of LSA to make hiddent topic be easily analyzed. LSA could not give topic labels, thus after examining the contents of each cluster researchers manually set topic labels.

The experiments were concerned about accuracy and processing time. For accuracy researchers analyzed threshold value of γ from Eq. 4 and saved the results into Table 1.

Value of γ has effect such that the smaller the value is the fewer the created cluster numbers are. For each value of γ we run 5 times experiments because there is random step in K-Means++ when selecting the next seed of centroid. In the end, for γ = 0.5 there are six clusters and for γ = 0.25 there are four clusters. This is

Table 1: Variation of γ value for clustering documents

	Accuracy level for experiment#					
γ	1	2	3	4	5	Avg.
0.25	0.64	0.76	0.71	0.88	0.81	0.76 [4 clusters]
0.50	0.86	0.80	0.77	0.83	0.82	0.82 [6 clusters]
0.75	0.70	0.89	0.41	0.73	0.55	0.66 [7 clusters]
1.00	0.08	0.31	0.59	0.51	0.38	0.37 [7 clusters]

Table 2: Results of clustering documents with their label of topics

Original topics		Clustered topics with K-means++, γ = 0.5	
Communication	15 docs	[#1] Communication/Surface	17 docs
Surface	15 docs		
Bioinformatics	15 docs	[#2] Bioinformatics 1	6 docs
		[#3] Bioinformatics 2	8 docs
Hyperspectral	15 docs	[#4] Hyperspectral 1	7 docs
		[#5] Hyperspectral 2	8 docs
		[#6] Unknown	14 docs

Table 3: Average times needed for plagiarism detection of 2 docs

Number of words	Without clustering (sec)	With clustering (sec)
≤1000	64	4
1001-1500	88	5
1501-2000	91	6

due to wide discussion topic on the cluster as a result, smaller clusters are required to accommodate the existing topics. Better accuracy has been achieved at γ = 0.5 with 0.82 as its accurateness value.

Topic labels for identified clusters are set manual. Table 2 shown identified clusters and comparison between the real topics and the clustered topics. Bioinformatics as one of hidden topics in Fig. 4 shows its wide discussion. Because of that the Bioinformatics topic in Table 2 is identified into two topics. Different from documents of cluster Communication and cluster Hyperspectral, documents that analyzed into cluster surface in Fig. 4 are scattered. This condition might lead documents within cluster surface into cluster unknown in Table 2.

Time required for detecting plagiarism on clustered documents was shorter than that on non clustered ones because the number of comparison needed has been reduced. Without clustering, the comparison between fingerprints of 60 documents in this experiment was ≥3000 times. Let assume that there are four clusters with the same number of document members which is 15 documents for each cluster. With clustering, we made four comparison first with cluster fingerprint. Then after selecting one closest cluster, researchers made the second comparison with ±15 fingerprints. Thus, all comparison needed was about ≤300 times which was less than comparison times without clustering. Table 3 shows the experiments measured by time needed for detecting similar texts. We grouped documents based on their word length, ±1000 terms and increased it by 500 terms (±1500 terms and ±2000 terms). Averagely time needed for comparison of document fingerprints could be reduced until about 10%. The experiments have shown that the approach with K-Means++ can make shorten process for detecting similar texts of clustered documents as a plagiarism indicator without sacrificing accuracy point. The approach of plagiarism detection includes usage of rule of thumb and Hartigan index to set standard value k for clustering with K-Means++.

CONCLUSION

In this study researchers conducted an approach for similar sentence detection as a plagiarism indicator. Based on previous researchs on Winnowing algorithm for comparing fingerprint document, we did fingerprint-based document clustering. The clustering process could reduce the comparison number needed for similar sentence detection. The Rule of Thumb and Hartigan Index were used to make sure that the number of cluster suitable for document clustering.

REFERENCES

Arthur, D., S. Vassilvitskii and K. Means, 2007. The advantages of careful seeding. Proceedings of the 18th Annual ACM-SIAM Symposium of Discrete Analysis, January 7-9, 2007, Louisiana, USA., pp: 1027-1035.

Butakov, S. and V. Scherbinin, 2009. The toolbox for local and global plagiarism detection. Comput. Educ., 52: 781-788.

Landauer, T.K., P.W. Foltz and D. Laham, 1998. An introduction to latent semantic analysis. Discourse Process., 25: 259-284.

Muth, R. and U. Manber, 1996. Approximate multiple strings search. Proceedings the 7th Annual Symposium on Combinatorial Pattern Matching, June 10-12, 1996, California, USA., pp: 75-86.

Oetsch, J., J. Puhrer, M. Schwengerer and H.Tompits, 2010. he system kato: Detecting cases of plagiarism for answer-set programs. Theory Pract. Logic Program., 10: 759-775.

Parapar, J. and A. Barreiro, 2009. Evaluation of text clustering algorithms with N-gram-based document fingerprints. Proceedings the 31st European Conference on Information Retrieval Research (ECIR), April 6-9, 2009, Toulouse, France, pp: 645-653.

Schleimer, S., D. Wilkerson and A.A. Winnowing, 2003. Local algorithms for document fingerprinting. Proceedings of the ACM Special Interest Group on Management of Data, June 9-12, 2003, San Diego, USA., pp: 76-85.

Wise, M.J., 1996. YAP3: Improved detection of similarities in computer program and other texts. Proceedings of the 27th ACM Special Interest Group on Computer Science Education, February 15-17, 1996, Philadelphia, USA., pp: 130-134.