Traceability Between Business Process and Software Component using Probabilistic Latent Semantic Analysis

Fony Revindasari[1], Riyanarto Sarno[2], Adhatus Solichah[3]
Informatics Department, Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
fony15@mhs.if.its.ac.id[1], riyanarto@if.its.ac.id[2], adhatus@if.its.ac.id[3]

***Abstract*—Business process and software component has relationship on business process execution in the organization or company. Changes in business process affecting the software component. A Method is needed to determine traceability of artifacts between process on business process and software component. The purpose of traceability is to trace the difference between business process of accompany and its software component through the artifacts. The artifacts in business process are identified by sequence of process, while the artifacts in software component are in the form of modules. In the proposed method, there are two main stage, namely modelling of process on business process and software component and document clustering using Probabilistic Latent Semantic Analysis (PLSA). In the modelling phase, process on business process and software component are grouped into documents. Then, the documents are processed by separating document into words. In the document clustering, documents are calculated using PLSA. It can be concluded that the document clustering can be done with recall 100% and precision 59%.**

Keywords—traceability, business process, software component, PLSA, cosine similarity.

I. INTRODUCTION

Business process is essential by organization and company to provide high quality of products and service [1]. It affects investment and income from the organization and company [2]. Business process has processes that relate to each other and each process has specific tasks [3]. In addition, business process is also linked with the software component on information techology in the organization or company [4]. Software component is supporting performance of business process. Changes in business process affects related software component or operating standards [5].

One way to identify relationship between business process and software component is identifying name of process and name of component. In reality, there is a difference in the name of business process and software component. However, if there is a different name then another way to explore the correlation between business process and software component is needed. In previous paper [6][7][8], some researcher have been tried to find correlation or similarity between business processes. However, in this paper, we focus on finding correlation between business process and software component.

A way to explore the correlation is finding traceability. Traceability determines trace between the different artifacts. In this case, the artifacts are process on business process and software component. Traceability is easy to do by clustering document of process on business process and software component. Previous study used tf idf method and cosine similarity to find the traceability. This method is easy to use but the result is not optimal to find traceability. So, in this proposed method, tf idf method changed to probabilistic latent semantic analysis to increase the accuracy.

Clustering document is a method for grouping objects into classes based on the similarity of these objects [9]. In the document clustering process, text is considered a vector that has elements with weighting based on frequency of words in the text called Words Space [10]. But, Word Space is not suitable for a large document. For a large document is required big dimensional vectors for word because of a lot of word frequency. So, Word Space is converted into concepts space to reduce the dimensional. Concept space is assumed the words that have same fequency in the same document has relationship so the words can be grouped into the topic. One method that can be used in concept space is Probabilistic Latent Semantic Analysis [11].

By using PLSA, the context of the document will be distinguished based on words with multiple meanings (disambiguate polysems) and grouped with words are the same or almost similar (synonyms) in its general context (topic) [11]. PLSA is also called statistical model (aspect model) to find patterns in text documents that it is easier to connect context with every word that appears on the document. With the modelling process, topic or context will be obtained from original text of the document without previous description of the document [12].

In PLSA, documents processes on business process and software components are included in the topics or certain context. But PLSA doesn't do similarity of words or keywords likes model of Latent Semantic Analysis (LSA) [11]. Similarities between topic and words in the documents can use

978-1-5090-1648-8/16/$31.00 ©2016 IEEE

similarity method Cosine Similarity [13]. Cosine similarity is used to search distance documents between process on business process and software component. The distance between documents process on business process and software component are calculated using probability value of the document to topic.

The paper is organized as follows: In Section 2, we review some of literature study of some previous researchers about document clustering. Dataset and methods that we propose contained in Section 3. In Section 4, we show the experimental process and experimental results. The following conclusions and future work is described in Section 5.

II. LITERATURE STUDY

Alignment of the relationship between process on business process and software components have been carried out by Aversano [4]. From his studies, it is known that there is a close relationship between process on business process and software component. Alignment process is performed by using the traceability matrix that can align business process and software component. But the data obtained are still bit ambiguous because of traceability source code and labeling of the name document business process and software component.

Earlier, Marcus [10] states that the search documentation in the source code is not suitable for all process on business process and software components. Incompatibility, it can impact on the analysis when the process of reverse engineering and maintenance when it will be reused. The solution offered is to conduct information retrieval method to be easy in maintenance using Latent Semantic Indexing (LSI). The results are quite promising but LSI can only ranked document based on top ranks.

In Pessiot [14] context of clustering documents is using unsupervised dimensional reduction has been proposed. The document is incorporated into the draft (topic words) by probabilistic topic. The same words from different documents became one topic. The words based on the number of occurances of words on the topic. Then, document will be included in the topic. PLSA is used as identification of topics and clustering documents into these topics.

Al-Anazi [15] has compared some method of clustering and similarity measurement method to increase value of cluster (k). Clustering method is used three clusters, namely k-means, k-means fast, and k-mediods. Similarity measurements is also used three methods, namely Cosine Similarity, Jaccard Similarity, and Correlation Coefficient.

III. METHODOLOGY

The proposed method is described as shown in Figure 1 as follows.

A. Preprocessing Data

The dataset used is an object oriented project. Description of processes on business process and software component should be modeled into processes document and software component document, respectively. Documents in the processes on business process are identified by the process name and documents in the software component is identified by the method name in each class.

After the documents are formed, then the preprocessing stage is performed as follows.

The initial stage is tokenization. Tokenization is to split documents into elements commonly called tokens. Next stage is stopword removal. This process begins with removal of all form of punctuation and removal of words that have no meaning or not important [16]. Usually, stopword removes connecting words and prepositions. Last stage is stemming. Stemming is the process of removing additive in a word that aims to obtain the basic form of the word [17]. In various documents, it can be found in various words actually comes from the same root, but written in different forms.

Having obtained a list of words from the preprocessing, the next process is to count the occurences of each word in each document which is used in the calculation of PLSA.

B. Probabilistic Latent Semantic Analysis

Probablistic Latent Semantic Analysis (PLSA) is used to calculate the probability of words and documents. PLSA can be used to identify words with multiple meanings and mapping those words in variety topics. Relationship between document, topic, and word can be seen in Figure 2.

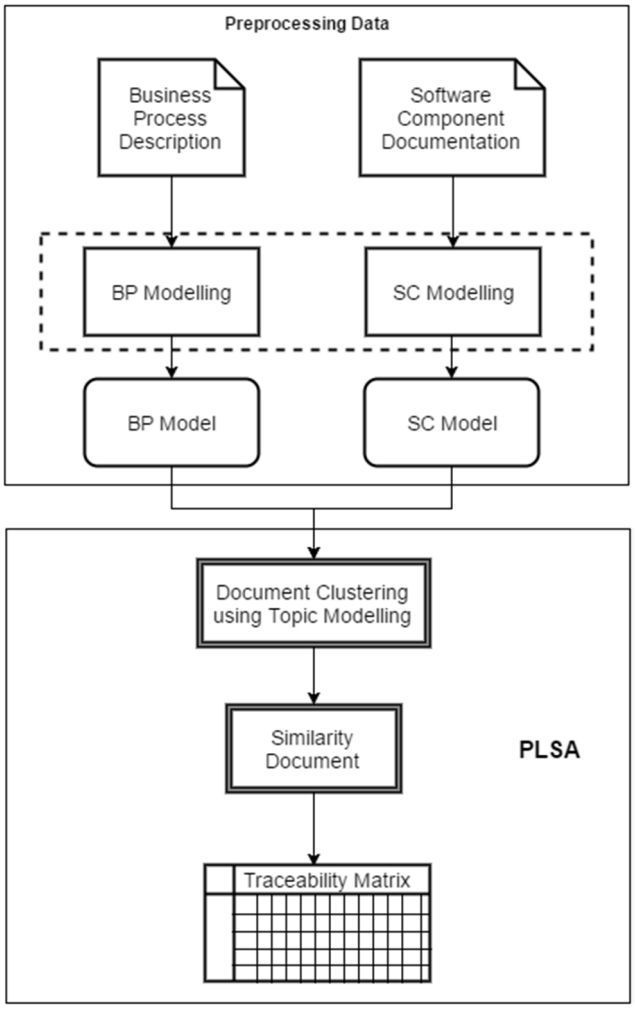

Fig. 1. Proposed method

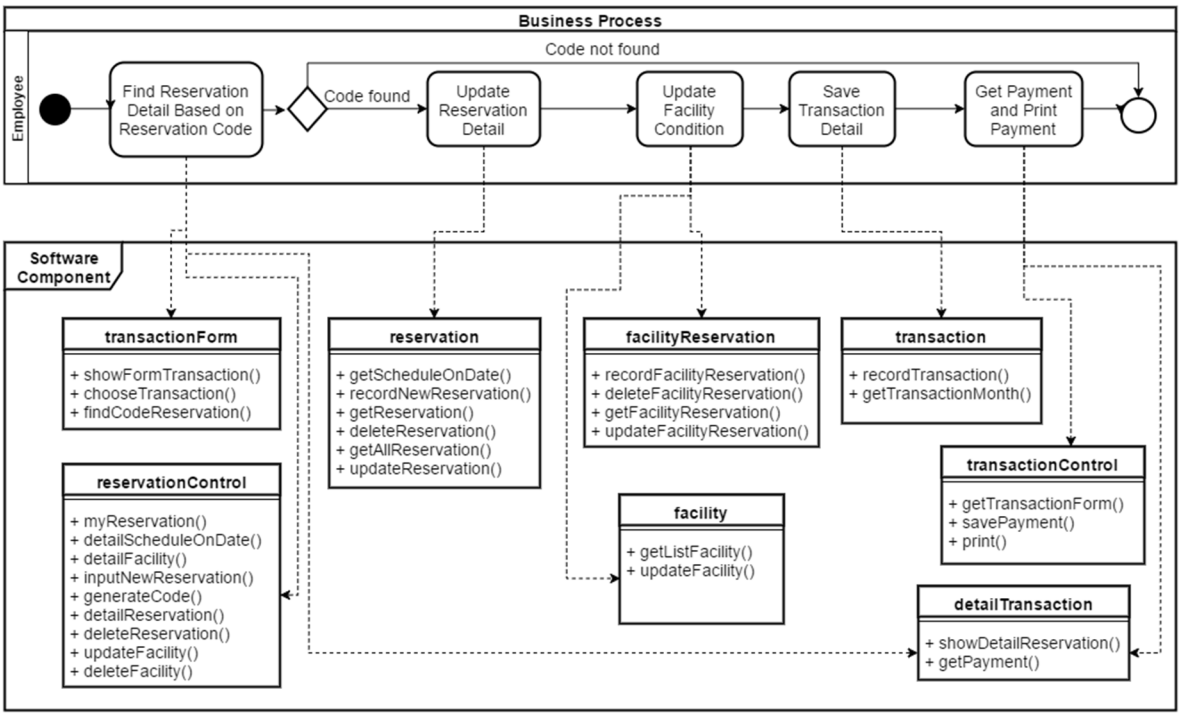

Fig. 2. Process on the business process and software component in sport facilities project

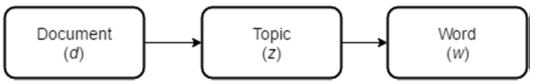

Fig. 3. Relationship among document, topic, and word

PLSA is usually used in applications of Information Retrieval or Natural Language Processing.

PLSA is used to classify words into topics that are not yet known (latent). So, each document is clustered based on topics. The algorithm is as follows : we determine the number of topics (z) then initialize parameters of probabilities : $P(z)$ is probability of topics, $P(d|z)$ is probability document that contains topic, $P(w|z)$ is probability of words contained in the topic are randomly. The calculation word in document is described in (1).

$$P(d_i, w_j) = \sum_{k=1}^{K} P(z_k)P(d_i|z_k)P(w_j|z_k) \tag{1}$$

The next step is to calculate the probability for each parameter using Expectation Maximization with two steps, namely E step and M step. E step is used to calculate the probability of the topics in the document and can be seen in (2).

$$P(z_k|d_i, w_j) = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{k=1}^{K} P(w_j|z_l)P(z_l|d_i)} \tag{2}$$

M step is used to renew the value of the parameter and can be seen in the (3) and (4).

The results of PLSA calculation are the probability of the word in a topic and the probability of topics in a document.

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i|w_j)P(z_k|d_i, w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(d_i|w_m)P(z_k|d_i, w_m)} \tag{3}$$

$$P(z_k|d_i) = \frac{\sum_{j=1}^{N} n(d_i|w_j)P(z_k|d_i, w_j)}{n(d_i)} \tag{4}$$

After calculating PLSA, the next step is to calculate the similarity between documents in process on business process and sofware component by using Cosine Similarity. Cosine similarity measures similarity between vectors of two documents. Vector A is probabilistic value probabilistic value of document business process in topic and vector B is probabilistic value of document software component in topic. Cosine similarity calculation can be seen from (5).

$$\cos(\theta) = \frac{A \,.\, B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\,\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5}$$

IV. Experimental Result

The dataset is used data from final project (object oriented programming). We choose the final project of object oriented programming that is sports facilities and described in Figure 3. Process on business process and software components are modeled into documents. There are 5 document of process on business process and 8 document of software component that is tested in this paper. The next step is preprocessing then we get list of term in each document.

PLSA calculation is using Java program to calculate each algorithm. The calculation steps as follows:

Step 1. Term that has been through a preprocessing stage, term made into a matrix by calculating the number of occurences of each term in a document.

Step 2. Determine the number of topics. In this case study, we determine 5 topics.

Step 3. Within each topic, there are 20 words (from document of processes on busines sprocess and software components) that are calculated based on the probability of a topic that is *P(z)*.

Step 4. Initialize *P(z|d)* probability of topic to document, for each topic to document randomly cummulative probability of topic to document $\sum(P(z|d)) = 1.0$.

Step 5. Initialize *P(w|z)* probability of term to topic, for each topic to document randomly cummulative probability of term to topic $\sum(P(w|z)) = 1.0$.

To enhance each value of topic probability, document to the topic probability, and term to the topic probability Expectation Maximization calculation is performed iteratively until convergent value is reached. Expectation Maximization calculation has two steps, namely E step and M step. E step is used to calculate the probability of the topics in the document. Iterations are used in E step to get the convergent value and to determine the threshold. Meanwhile, M step is used to renew value of parameters.

The result of these PLSA calculations are the probability of the word in a topic and the probability of the topic within a document. The result of these calculation are presented in Table 1 and Table 2.

From these table can be seen the results of the probability of each topic in document 1,2 to 13. In Table 1, it can be concluded that the probability value of 1 or close to 1 indicates that the topic is compatible for document process on business process or document software component.

TABLE I. PROBABILITY DOCUMENT IN TOPIC

Document	Topic				
	Topic 0	***Topic 1***	***Topic 2***	***Topic 3***	***Topic 4***
Doc P 1	1	4.69E-55	0	9.99E-38	0
Doc P 2	0.75	5.45E-38	0	1.30E-44	0.25
Doc P 3	3.94E-42	1.12E-108	0.150608	0.849392	1.44E-61
Doc P 4	0	0.427637	2.44E-143	0.572363	3.03E-17
Doc P 5	0	0.427637	4.68E-152	0.572363	4.43E-19
Doc SC 1	0.666667	2.61E-99	0.333333	4.79E-14	0
Doc SC 2	0	0	1	8.93E-61	0
Doc SC 3	0.333333	0.666667	0	0	0
Doc SC 4	1.12E-45	0	0.857143	8.90E-14	0.142857
Doc SC 5	3.04E-04	0.999696	0	3.53E-17	0
Doc SC 6	0.223329	9.40E-132	3.13E-08	0.776671	0
Doc SC 7	4.55E-34	5.57E-103	1.19E-31	1	1.08E-62
Doc SC 8	1.43E-40	0	0.857143	2.32E-14	0.142857

TABLE II. PROBABILITY TERM IN TOPIC

Term	Topic				
	Topic 0	*Topic 1*	*Topic 2*	*Topic 3*	*Topic 4*
detail	2.10E-06	0.07459	5.67E-56	0.037292	2.10E-06
reservation	1.12E-05	0.102917	0.22423	0.378899	1.12E-05
find	8.06E-07	2.04E-05	0.204068	0.041873	8.06E-07
based	1.27E-204	3.32E-61	2.75E-59	0.03728	1.27E-204
input	9.56E-11	0.149231	1.19E-22	1.07E-18	9.56E-11
generate	2.01E-14	0.298462	7.68E-26	3.05E-23	2.01E-14
delete	0.100228	2.84E-58	1.57E-10	0.07456	0.100228
date	2.16E-67	2.44E-96	0.129729	0.233532	2.16E-67
schedule	2.80E-30	8.31E-60	0.08746	0.055228	2.80E-30
update	8.74E-33	4.84E-63	0.111432	0.104053	8.74E-33
facility	0.599266	0.00156	7.33E-22	2.80E-55	0.599266
show	0.100227	4.50E-244	0	2.01E-217	0.100227
save	5.75E-05	0.149188	2.52E-93	1.30E-56	5.75E-05
payment	0.200202	1.88E-04	2.73E-13	1.29E-96	0.200202
transaction	5.45E-09	0.149231	6.73E-262	0	5.45E-09
choose	0	0	0.034726	4.04E-29	0
form	5.30E-06	0.074612	4.81E-106	7.88E-52	5.30E-06
record	1.02E-25	9.16E-56	0.069452	2.46E-07	1.02E-25
month	1.45E-25	8.93E-57	0.069451	7.87E-07	1.45E-25
print	1.21E-152	1.25E-99	0.034726	1.96E-18	1.21E-152
detail	2.73E-154	2.69E-100	0.034726	1.43E-18	2.73E-154
reservation	0	0	1.09E-22	0.03728	0

In Table 2, it can be concluded that each term has a probability topics. So, the probability value closes to 1, these terms are grouped in the topic.

By using probability of topic, we can calculate the similarity between documents using Cosine Similarity. The result of calculation using cosine similarity can be seen in Table 3.

After cosine similarity calculation is complete, the next step is to descibe the traceability matrix. Traceability matrix is used to determine the trace between processes on business process and software component. Traceability matrix is shown in Table 4. The 'x' indicates that the link retrieved does not match with the relevant value. The 'o' indicates that the link retrieved is correctly and relevant.

$$Recall = \frac{\sum_i \#(Relevant_i \cap Retrieved_i)}{\sum_i Relevant_i} \%$$
$$Precision = \frac{\sum_i \#(Relevant_i \cap Retrieved_i)}{\sum_i Retrieved_i} \% \quad (6)$$

The result of cosine similarity calculations are used to calculate the value of recall and precision. Before calculating the value of recall and precision, the value of threshold must be determined to give similarity value limits. The value of threshold is obtained from observation when performed experiments. It threshold value is 0.35 from previous observation in probability calculation. If the similarity value is above the threshold value, the data is considered be the same. The calculation of recall and precision adopted by using [4] can be seen in (6). The result of recall and precision calculation can be seen in Table 5.

TABLE III. COSINE SIMILARITY CALCULATION

Process Name	Software Component							
	SC 1	SC 2	SC 3	SC 4	SC 5	SC 6	SC 7	SC 8
find reservation detail based on reservation code	0.804747	0.447214	0.239466	0	0.447209	3.18E-04	1.09E-31	1.89E-74
update reservation detail	8.11E-05	2.25E-04	0.424264	0.569042	1.24E-42	7.48E-14	9.05E-28	0.1
update facility condition	1.09E-31	1.89E-74	2.48E-22	0.985138	0.239466	0.971416	3.20E-19	7.29E-13
save transaction detail	1.59E-64	1.83E-124	0.894427	0	5.22E-54	0.424264	0.928477	1.07E-140
get payment and print receipt	6.33E-12	1.87E-82	8.27E-134	2.41E-126	0.772539	0.215366	1.27E-104	0.807842

TABLE IV. TRACEABILITY MATRIX

Process on the Business Process	Software Component							
	SC 1	SC 2	SC 3	SC 4	SC 5	SC 6	SC 7	SC 8
P 1	o	o	x		o			
P 2			o	x				
P 3				o	x	o		
P 4			o			o	x	
P 5					o	x		o

TABLE V. RECALL AND PRECISION VALUE (TOTAL VALUE)

Process Name	Relevant	Retrieved	Relevant and Retrieved	Precision	Recall
find reservation detail based on reservation code	3	4	3	75%	100%
update reservation detail	1	2	1	50%	100%
update facility condition	2	3	2	67%	100%
save transaction detail	1	3	1	33%	100%
get payment and print receipt	2	3	2	67%	100%
Total				59%	100%

Based on Table 4, the value of precision is obtained low precision because there is an error in getting the retrieved value. But, this precision higher than previous paper [4]. This is because the similarity calculation using the probability topic value is affected by the determination the of topic and iterations in the PLSA calculation.

V. CONCLUSION

In this paper, we have performed traceability artifacts on process on the business process and software component. The result of the traceability artifact is used to trace between the different artifacts. Traceability of process on the business process and software component is using document clustering. Document clustering between process on business process and software component are used to cluster document into contents or topics. The document clustering is used Probabilistic Latent Semantic Analysis (PLSA). PLSA is used to get value of the probability of topic, documents on the topic, and term on the topic. But, calculation by using PLSA can not figure out the similarity between document of process on business process and software component. So, similarity calculation is used Cosine Similarity. Cosine similarity is to determine the similarity between two vectors. The vectors are process on business process and software component. Input for calculation is value of probability document on the topic.

For the result, PLSA method can increase the accuracy rather than tf idf method in previous study. The value of precision is obtained low precision because there is an error in getting the retrieved value. This is because the similarity calculation using the probability topic value is affected by determining the topic and iterations in the PLSA calculation. Furthermore, the contents and the number of the document also affect the value of the precision of PLSA calculation.

For future work, dataset used large scope and huge content of process on business process and software component. The dataset should not limited to the name of the process on business process and software component. It should be added the source code from software component. So, the value of similarity is higher than the value of process name and software component name**.**

REFERENCES

[1] A. Tarhan, O. Turetken, and H. A. Reijers, "Business process maturity models: A systematic literature review," *Information Software Technology*, vol. 75, pp. 122–134, 2016.

[2] W. Bandara, M. Indulska, S. Chong, and S. Sadiq, "Major Issues in Business Process Management: An Expert Perspective," *ECIS 2007 - 15th European Conference on Information System*, vol. 2007, pp. 1240–1251, 2007.

[3] M. Von Rosing, H. Von Scheel, and A. W. Scheer, *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM*, vol. 1. 2014.

[4] L. Aversano, C. Grasso, and M. Tortorella, "Managing the alignment between business processes and software systems," *Information Software Technology*, vol. 72, pp. 171–188, 2016.

[5] R. Sarno, H. Ginardi, E. W. Pamungkas, D. Sunaryono "Clustering of ERP business process fragments," *International Conference on Computer, Control, Informatics and Its Applications (IC3INA),* pp. 319-324, 2013.

[6] R. Sarno, E. W. Pamungkas, D. Sunaryono, and Sarwosri, "Business process composition based on meta models," *2015 International Seminar on Intelligent Technology and Its Application ISITIA 2015 - Proceeding*, pp. 315–318, 2015.

[7] Z. Yan, R. Dijkman, and P. Grefen, "Fast business process similarity search with feature-based similarity estimation," *Lecture Notes on Computer Science (including Subser. Lecture Notes Artificial Intelligent Lecture Notes Bioinformatics)*, vol. 6426 LNCS, no. PART 1, pp. 60–77, 2010.

[8] M. Ehrig, “Measuring Similarity between Business Process Models.pdf.”

[9] R. Dijkman, M. Dumas, B. Van Dongen, R. Krik, and J. Mendling, “Similarity of business process models: Metrics and evaluation,” *Information System*, vol. 36, no. 2, pp. 498–516, 2011.

[10] a. Marcus and J. I. Maletic, “Recovering documentation-to-source-code traceability links using latent semantic indexing,” *25th International Conference on Software Engineering, 2003. Proceedings.*, vol. 6, pp. 125–135, 2003.

[11] T. Hofmann, “Unsupervised learning by probabilistic Latent Semantic Analysis,” *Machine Learning*, vol. 42, no. 1–2, pp. 177–196, 2001.

[12] D. Blei, L. Carin, and D. Dunson, “Probabilistic topic models,” *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 55–65, 2010.

[13] L. Yuanchao, W. Xiaolong, X. Zhiming, and G. Yi, “A Survey of Document Clustering,” 2006.

[14] J. F. Pessiot, Y. M. Kim, M. R. Amini, and P. Gallinari, “Improving document clustering in a learned concept space,” *Information Processing Managing*, vol. 46, no. 2, pp. 180–192, 2010.

[15] S. Al-Anazi, H. AlMahmoud, and I. Al-Turaiki, “Finding Similar Documents Using Different Clustering Techniques,” *Procedia Computer Science*, vol. 82, no. March, pp. 28–34, 2016.

[16] T. Verma, “Tokenization and Filtering Process in RapidMiner,” *International Journal Application Information System – ISSN 2249-0868 Found. Computer Science FCS, New York, USA*, vol. 7, no. 2, pp. 16–18, 2014.

[17] S. Ferilli, F. Esposito, and D. Grieco, “Automatic learning of linguistic resources for stopword removal and stemming from text,” *Procedia Computer Science*, vol. 38, no. C, pp. 116–123, 2014.