Advances in Knowledge-Based and Intelligent Information and Engineering Systems
M. Graña et al. (Eds.)
IOS Press, 2012

doi:10.3233/978-1-61499-105-2-595

An Algorithm for Tolerance Value Generator in Tolerance Rough Sets Model

Gloria Virginia and Hung Son Nguyen

University of Warsaw, Faculty of Mathematics, Informatics and Mechanics
Banacha 2, 02-097 Warsaw, Poland

Abstract. The Tolerance Rough Sets Model (TRSM) is a tool to model document in text mining. Generation of a document representation based on TRSM basically depends on tolerance classes of terms which are created by setting up a tolerance value. Despite the fact that tolerance value is critical in TRSM, the manual process of setting this value is an exhaustive task. We conducted a study on our own corpus and were supported by two human experts when constructing the training data. We came up with a novel algorithm to generate tolerance value automatically from a set of training data. The heart of our algorithm is measuring the distance between document representation calculated using TFIDF weighting scheme and document representation yielded by TRSM both in reduced dimensional space generated by Singular Value Decomposition (SVD). In spite of the result, we recognize that further study is significant, i.e. to set up some properties (the size of training data and rank of SVD) as well as evaluating the algorithm in real data and scenario.

Keywords: Tolerance rough sets model, text mining, singular value decomposition

1 Introduction

The Tolerance rough set model (TRSM) is a method introduced by Kawasaki, Nguyen, and Ho [3] in 2000 to construct a document representation in text mining for the task such as information retrieval or clustering. It has been shown in [4], [7], and [2] that the document representations yielded by TRSM (TRSM-representation) were richer than the one based on TFIDF weighting scheme (TFIDF-representation), and brought better results for the tasks given.

There are three main components of TRSM for work which are dependent in sequence: *a*) a tolerance matrix, that consists of tolerance classes of all terms in the document collection; *b*) the upper document representation, that represents the occurrence of terms in a document based on the upper set of a document; and *c*) the extended weighting scheme (TRSM weighting scheme). The TRSM-representation can be seen as the revised version of TFIDF-representation which is recalculated using the TRSM weighting scheme. During calculation, it considers the upper document representation which is created based on tolerance matrix. It is the nature of TRSM to depend on a *tolerance co-occurrence* value, θ,

of two terms in document collection when constructing the term-by-term matrix where the θ value represents the importance of relationships between terms.

Despite the fact, that the value of θ is crucial for TRSM implementation, there is no consensus about how we can set a certain number of θ. It is usually chosen by the researcher or human expert based on manual inspection through the training data or his/her consideration about the data. It is not deniable that each datum is distinctive hence requires different treatment, however, determining the θ value by hand is an exhaustive task before even starting the TRSM paths.

In this paper, we present an algorithm to generate a tolerance value θ automatically based on the training data. We implemented a cognitive approach in terms that we learned from human and tried to mimic the result yielded by the human thinking process. We employed the singular value decomposition (SVD) in order to create a lower n-dimensional vector of a document. The heart of the algorithm is measuring the distance between two reduced document representations in a certain range of θ candidates. The contributions of this paper are twofold: *a*) we introduce a novel algorithm for an automatic tolerance value generator; and *b*) we verify the learning process of θ determination. Finally, we point out some directions for further study with regard to TRSM implementation.

2 Background of Tolerance Rough Sets Model

In order to generate a richer document representation in terms of semantic relatedness, TRSM needs to create tolerance classes of terms and approximations of subsets of documents. If $D = \{d_1, d_2, ..., d_N\}$ is a set of documents and $T = \{t_1, t_2, ..., t_M\}$ is a set of index terms from D, then the tolerance classes of terms in T are created based on the co-occurrence of index terms in all documents from D. A term weight vector is used to represent each document $d_i = \{w_{i,1}, w_{i,2}, ..., w_{i,M}\}$, where $w_{i,j}$ is the weight of term t_j in document d_i.

Rough sets theory [5], which is the root of TRSM, says that any concept can be approximated by its lower and upper approximations, and the vagueness of the concept is defined by the region between its upper and lower approximations. In information retrieval context, we can assume a term as a concept. Thus, implementing TRSM means that we approximate concepts determined over the set of terms T on a tolerance approximation space $\mathcal{R} = (T, I_\theta, \nu, P)$ by employing tolerance relation; where I_θ is an uncertainty function, ν is a vague inclusion function, and P is a structural function. The following are their definitions.

- **Uncertainty function:** tolerance relation requires two properties, which are reflexive (xRx) and symmetric ($xRy \rightarrow yRx$). Thus the **tolerance class** of term t_i is defined as

$$I_\theta(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \theta\} \cup \{t_i\} \ . \tag{1}$$

  where θ is a positive parameter and $f_D(t_i, t_j)$ denotes the number of documents in D where both terms t_i and t_j appear.

- **Vague inclusion function:** the vague inclusion ν is defined as $\nu(X,Y) = \frac{|X \cap Y|}{|X|}$, thus the membership function μ for $t_i \in T, X \subseteq T$ is defined as

$$\mu(t_i, X) = \nu(I_\theta(t_i), X) = \frac{|I_\theta(t_i) \cap X|}{|I_\theta(t_i)|} \tag{2}$$

- **Structural function:** all tolerance classes of index terms are considered as structural subsets, hence $P(I_\theta(t_i)) = 1$ is for all $t_i \in T$.

With these definitions, we can define the lower approximation, upper approximation, and boundary region of any subset $X \subseteq T$ in tolerance space $\mathcal{R} = (T, I, \nu, P)$ as $L_\mathcal{R}(X) = \{t_i \in T \mid \nu(I_\theta(t_i), X) = 1\}$, $U_\mathcal{R}(X) = \{t_i \in T \mid \nu(I_\theta(t_i), X) > 0\}$, and $BN_\mathcal{R}(X) = U_\mathcal{R}(X) - L_\mathcal{R}(X)$ respectively.

The richer representation of document $d_i \in D$ is achieved by representing document with its upper approximation, i.e.

$$\mathbf{U}_\mathcal{R}(d_i) = \{t_i \in T \mid \nu(I_\theta(t_i), d_i) > 0\} \tag{3}$$

followed by calculating the weight vector using an extended weighting scheme, i.e.

$$w^*_{ij} = \frac{1}{S} \begin{cases} (1 + \log f_{d_i}(t_j)) \log \frac{N}{f_D(t_j)} & \text{if } t_j \in d_i \\ \min_{t_k \in d_i} w_{ik} \frac{\log \frac{N}{f_D(t_j)}}{1 + \log \frac{N}{f_D(t_j)}} & \text{otherwise} \end{cases} \tag{4}$$

where S is a normalization factor. The extended weighting scheme is defined from the standard TFIDF weighting scheme and is necessary in order to handle terms that occur in a document's upper approximation but, not in the document itself.

By employing TRSM, the final document representation has less zero valued similarity, that leads to higher possibility of two documents have non-zero similarities although they do not share any terms. This is the main advantage of TRSM-based algorithm claimed to have over traditional approaches.

3 Methodology

We worked with two choral experts intensively in an annotation process in order to construct the ground truth of this study. The annotation process consisted of two tasks, which were *a*) topic assignment, where the human experts assigned topic(s) for each document within the original corpus; and *b*) keywords determination, where they determined terms considered as highly related with the topic(s) given. This process aimed to grasp how the topic(s) could be assigned to a particular document that was mainly described by the keywords determined. In this study, we took benefit from these keywords as the list of essential terms related to the topic of the document, i.e. the document, and assumed that the other terms not listed were less significant terms. The topic assignment yielded 127 topics and the keywords determination yielded a new corpus, called WORDS-corpus.

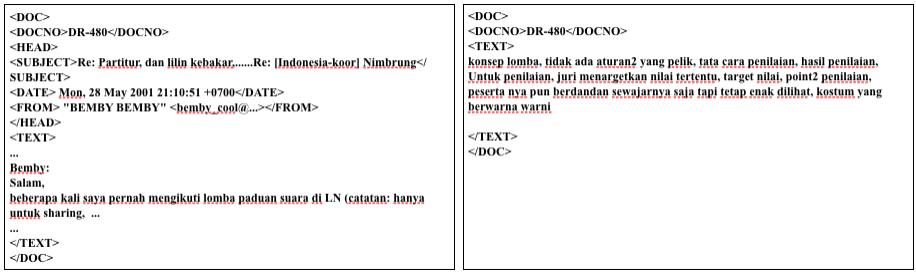

Fig. 1. The content of corpora. Picture on the left is an example of ICL-corpus document which consists of original document. Picture on the right is an example of WORDS-corpus document which consists of keywords given by human expert for particular ICL-corpus document, i.e. the ICL-corpus document with number "DR-480" shown on the left.

The main corpus of this study, called ICL-corpus, consists of 1,000 first emails of Indonesian Choral Lovers (ICL) Yahoo! Groups and was formatted as of the Text REtrieval Conference (TREC) format [8]. Therefore, the test collections consist of three parts (a set of documents, a set of information needs, and a set of relevance judgments) and all documents are marked up in a TREC-like format. Consult Fig. 1 to see the content of both corpora. Notice that the main difference between them lies in the *text body* of document, i.e. the document of ICL-corpus consists of a body of email while the document of WORDS-corpus consists of keywords defined by human experts.

4 Experiment

Preprocessing Phase: We preprocessed the corpora by taking documents within both corpora as inputs and came up with the TFIDF-representations for each corpus. We implemented an information retrieval library freely available called Lucene[1] with some modifications in order to implement a version of stopword list and stemmer specifically for Indonesian language, called Vega's stopword [6] and Confix-Stripping stemmer (CS-stemmer) [1] respectively.

TRSM Phase: We implemented the tolerance rough sets model in this phase, which means we converted the TFIDF-representation into TRSM-representation by following these steps for both corpora:

1. Construct a tolerance matrix comprises tolerance classes of all terms based on (1). For the purpose of this study, we altered θ value from 0 to 25.
2. Create upper approximation of documents $U_{\mathcal{R}}(d_i)$ using (3).
3. Generate TRSM-representations by recalculating the TFIDF-representations using (4) and considering the upper approximation of documents $U_{\mathcal{R}}(d_i)$.

[1] http://lucene.apache.org/.

SVD Phase: The objective of this phase was reducing the dimensionality space of document representation so it could be plotted on low dimensional graph, and further analyzed. We calculated the SVD 2-rank and 10-rank over TFIDF-representation of ICL-corpus (ICL-TFIDF-representation) and TRSM-representation of ICL-corpus and WORDS-corpus (ICL-TRSM-representation and WORDS-TRSM-representation respectively), using the SVD algorithm. In this experiment, we employed a Java package called JAMA[2] to do the job.

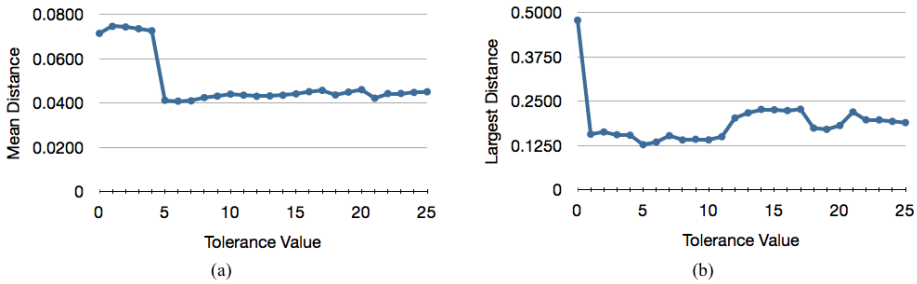

Fig. 2. The distance between ICL-TRSM-representation and WORDS-TRSM-representation for SVD 2-rank where $0 \leq \theta \leq 25$ based on (a) mean distance and (b) largest distance.

Evaluation Phase: In the evaluation phase, we did two tasks:

1. Calculating the mean average distance and the largest distance between (a) TRSM-representations of both corpora and (b) TFIDF-representation and TRSM-representation of the ICL-corpus.
2. Plotting the SVD-representations we had in 2 dimensional space of ICL-TFIDF-representation, ICL-TRSM-representation, and WORDS-TRSM-representation.

For distance calculation, we used the Euclidean distance function

$$d(V,U) = \sqrt{\sum_{i=0}^{M}(v_i - u_i)^2},$$

where $[v_i]_{i=0}^{M}$ and $[u_i]_{i=0}^{M}$ denote weight vectors of document V and U.

5 Result and Discussion

5.1 Learning from WORDS-corpus

We take an assumption that each document of WORDS-corpus consists of essential keywords, which should occur in particular document representation of

[2] http://math.nist.gov/javanumerics/jama/.

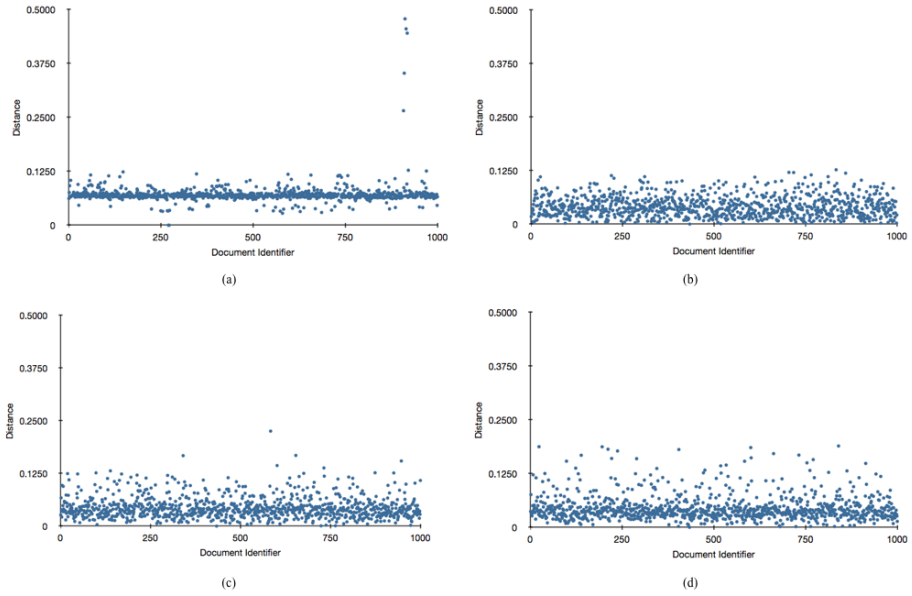

Fig. 3. The scatter graph of distance between documents within ICL-corpus and WORDS-corpus when (a) $\theta = 0$, (b) $\theta = 5$, (c) $\theta = 15$, and (d) $\theta = 25$.

ICL-corpus. Therefore, the distance between document representations of both corpora measures how far ICL-corpus from WORDS-corpus. It brings us to preference of lower value of distance.

Figure 2(a) depicts the mean average distance between TRSM-representation of ICL-corpus and WORDS-corpus after they were reduced into 2 dimensions for tolerance value 0 to 25. Figure 2(b) is a graph of the largest distance of similar calculation. With regard to the notion that θ is a threshold for filtering out the terms by its co-occurrence within documents in a corpus, Fig. 2 seems to give us a clue that the way TRSM remove unimportant terms in documents, i.e. by filtering out some terms on specific co-occurrence value, is adequate in order to make documents of ICL-corpus closer to documents of WORDS-corpus.

Consider Fig. 3 that depicts the scatter graph of distance between ICL-corpus and WORDS-corpus for tolerance value 0, 5, 15, and 25. From these figures, it is clear that when θ is getting higher, the graph becomes more sparse, and the distance between documents is closer to 0. However, the distance tends to be far away from 0 if the θ is set up too high.

Figure 4 shows the results of distance calculation similar to Fig. 2 unless it is for SVD 10-rank, i.e. Fig. 4(a) is for mean distance and Fig. 4(b) is for largest distance. From both graphs, we see that the mean distance and the largest distance are walking in line, i.e. significant changes are shown on $\theta = 5$ and no other significant change occurs after that value.

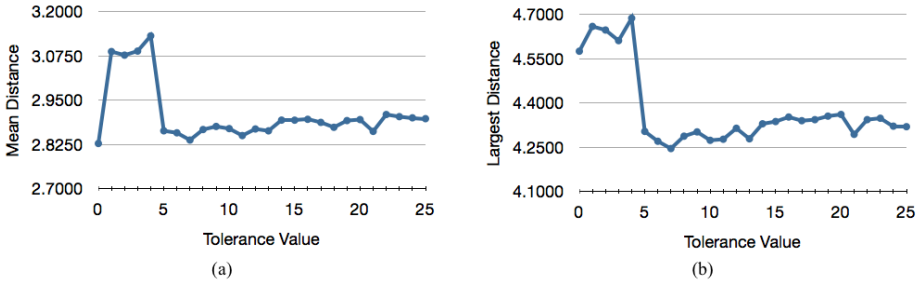

Fig. 4. The distance between ICL-TRSM-representation and WORDS-TRSM-representation for SVD 10-rank where $0 \leq \theta \leq 25$ based on (a) mean distance and (b) largest distance.

We can see that graphs of the mean distance and the largest distance are more similar in SVD 10-rank than in SVD 2-rank. Furthermore, regression analysis[3] of both distances for SVD 2-rank as well as SVD 10-rank shows that statistically the distance value tends to be smaller for larger tolerance value. Therefore, we suggest a rank value as close as 10 for SVD. However from an efficiency perspective, calculating distance of 10 dimensional vectors is more expensive than calculating distance of 2 dimensional vectors. Thus, this trade-off should be considered while deciding a rank for SVD.

5.2 Experiment on ICL-corpus

In this section, we present and discuss results of comparison between TFIDF-representation and TRSM-representation on singular corpus only, i.e. ICL-corpus, after dimensionality of those representations were reduced into 2 and 10 using SVD method. Based on the capability of TRSM which is to enrich a document representation, larger distance is preferred on this section because it gives us indication that TRSM is enricher the original document representation, i.e. the TFIDF-representation.

Graphs in Fig. 5(a) and 5(b) depict results of distance calculation based on mean distance and largest distance respectively on SVD 2-rank. For largest distance, there is no considerable change resulted, however, it seems that the TRSM-representations are closer each other when θ value is between 5 to 11. This phenomenon leads to high mean distances as suspected and is confirmed by Fig. 5(a). The highest mean distance happens on $\theta = 5$.

The results of distance calculation for SVD 10-rank presented on Fig. 6(a) and Fig. 6(b) for mean distance and largest distance sequentially. It is clear from these graphs that the distance value is getting better for higher tolerance value, and the highest mean distance happens on $\theta = 21$. Learn from the regression analysis, we found that statistically the distance value tends to be larger for

[3] Regression analysis was computed by Wolfram Mathematica.

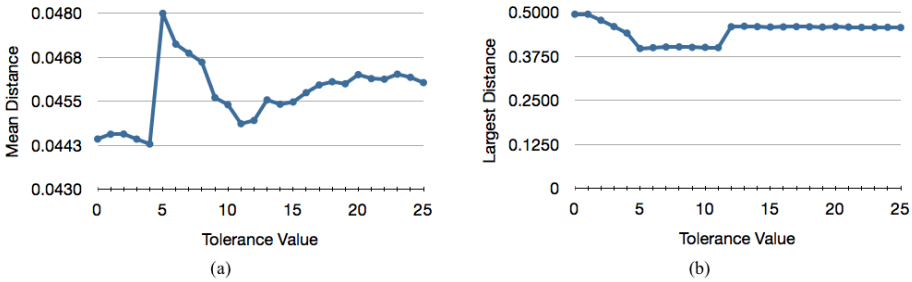

Fig. 5. The distance between ICL-TFIDF-representation and ICL-TRSM-representation for SVD 2-rank where $0 \leq \theta \leq 25$ based on (a) mean distance and (b) largest distance.

higher tolerance value and the slope of the regression line based on SVD 10-rank is sharper than on SVD 2-rank. By considering these data, our suggestion about SVD's rank is the same, i.e. we urge to choose a value for rank of SVD as close as 10.

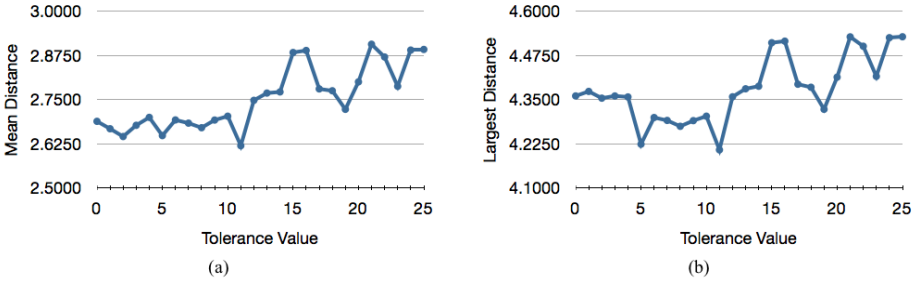

Fig. 6. The distance between ICL-TFIDF-representation and ICL-TRSM-representation for SVD 10-rank where $0 \leq \theta \leq 25$ based on (a) mean distance and (b) largest distance.

5.3 Algorithm for Tolerance Value Generator

Based on the results presented in previous sections, we found that we can take advantage from WORDS-corpus, as training data, and pick θ value of the smallest mean distance between TRSM-representation of ICL-corpus and WORDS-corpus over a range of θ value candidates. If there is no such WORDS-corpus available, it seems that we can still take advantage from a single corpus, i.e. create a set of training data from it, and pick θ value of the highest mean distance between its TFIDF-representation and its TRSM-representation over a range of θ value candidates.

We acknowledge that a training data as WORDS-corpus is hardly to create, hence the proposed algorithm uses a single corpus. Finally, the algorithm to generate a tolerance value θ automatically is as follows

Input: A set of documents as training data.
Output: A tolerance value.
Process:

1: construct TFIDF-representation
2: **for** $lower\_bound \leq \theta \leq upper\_bound$ **do**
3: construct TRSM-representation
4: calculate SVD for TFIDF-representation and TRSM-representation
5: calculate mean distance between SVD-representations
6: **end for**
7: select θ with the highest mean distance

We consider that it is too early for setting the upper bound (and also the lower bound) of tolerance value to be used in the algorithm. However, for now, we urge to choose $\theta > 4$ for the lower bound due to the fact that significant change occurs on this value, and in order to provide larger flexibility for SVD up to 10-rank, let us set the upper bound as 22. Thus, we may use this range of tolerance value $5 \leq \theta \leq 22$ in the algorithm.

Recall Fig. 2 and 4 on section 5.1, we can see that the distance value on $\theta = 21$ is close enough to the best distance value compared with the other value on the rest of θ. In [7], $\theta = 21$ was suggested based on manual inspection on the results yielded by altering the tolerance value from 0 to 25. This value came up based on the high percentage of co-occurrence terms between index term of WORDS-corpus and index term in the upper set of topic within ICL-corpus (97.58%), and on this θ value, the number of index term in the upper set of topic within ICL-corpus (which showed to be large) could be reduced significantly. Thus, it seems that we may expect effective TRSM-representations by employing SVD 10-rank.

The time complexity of the algorithm is $O(N^2M^2K)$, where N is the number of documents, M is the number of index term, and K is the number of θ values being evaluated. Therefore, a small number of training data are preferred.

6 Conclusion

Tolerance value θ is a crucial value that has to be set up prior to the implementation of TRSM. We did a study using two corpora and came up with a novel algorithm to generate a tolerance value automatically using a set of training data. We learned from WORDS-corpus that we might choose a tolerance value θ whose mean distance was the highest. The mean distance is calculated between the TFIDF-representation and TRSM-representation of the training set in their low dimensional space which are constructed using SVD over a range of θ values.

Some properties of the algorithm have been discussed, and some values were suggested, i.e. lower bound is $\theta \geq 5$, upper bound is $\theta \leq 22$, and SVD rank is

10. Due to its infancy, it is necessary to conduct further study regarding with some parameters, e.g. the size of training data and the rank of SVD. We had verified the learning process to set the θ value by employing the WORDS-corpus, however, we still need to validate it by implementing the algorithm into a more realistic scenario, i.e. information retrieval and clustering. Furthermore, it is tempting to validate the algorithm using the standard corpus, e.g. TREC.

Acknowledgments. This work is partially supported by Specific Grant Agreement Number-2008-4950/001-001-MUN-EWC from European Union Erasmus Mundus "External Cooperation Window" EMMA, and the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic Scientific Research and Experimental Development Program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information" and grants from Ministry of Science and Higher Education of the Republic of Poland N N516 077837. We thank Faculty of Computer Science, University of Indonesia, for the permission of using the CS stemmer.

References

1. Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S.M., Williams, H.E.: Stemming indonesian: A confix-stripping approach 6, 1–33 (December 2007), http://doi.acm.org/10.1145/1316457.1316459
2. Gaoxiang, Y., Heping, H., Zhengding, L., Ruixuan, L.: A novel web query automatic expansion based on rough set. Wuhan University Journal of Natural Sciences 11(5), 1167–1171 (2006)
3. Kawasaki, S., Nguyen, N.B., Ho, T.B.: Hierarchical document clustering based on tolerance rough set model. In: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery. pp. 458–463. PKDD '00, Springer-Verlag, London, UK (2000), http://dl.acm.org/citation.cfm?id=645804.669983
4. Nguyen, H.S., Ho, T.B.: Rough Document Clustering and the Internet, chap. 47, pp. 987–1003. John Wiley & Sons Ltd. (2008)
5. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11(5), 341–356 (1982)
6. Vega, V.B.: Information Retrieval for the Indonesian Language. Master's thesis, National University of Singapore (2001), unpublished
7. Virginia, G., Nguyen, H.S.: Investigating the effectiveness of thesaurus generated using tolerance rough set model. In: Proceedings of the 19th International Conference on Foundations of intelligent systems. pp. 705–714. ISMIS'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2029759.2029848
8. Voorhees, E.M., Harman, D.: Overview of the ninth text retrieval conference (trec-9). In: Proceedings of the Ninth Text REtrieval Conference (TREC-9. pp. 1–14. National Institute of Standards and Technology (NIST) (2000), http://trec.nist.gov/pubs/trec9/papers/overview_9.pdf