Fundamenta Informaticae 124 (2013) 27–46
DOI 10.3233/FI-2012-823
IOS Press

Lexicon-based Document Representation

Gloria Virginia[*]
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
virginia@icm.edu.pl

Hung Son Nguyen
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
son@mimuw.edu.pl

Abstract. It is a big challenge for an information retrieval system (IRS) to interpret the queries made by users, particularly because the common form of query consists of very few terms. Tolerance rough sets models (TRSM), as an extension of rough sets theory, have demonstrated their ability to enrich document representation in terms of semantic relatedness. However, system efficiency is at stake because the weight vector created by TRSM (TRSM-representation) is much less sparse. We mapped the terms occurring in TRSM-representation to terms in the lexicon, hence the final representation of a document was a weight vector consisting only of terms that occurred in the lexicon (LEX-representation). The LEX-representation can be viewed as a compact form of TRSM-representation in a lower dimensional space and eliminates all informal terms previously occurring in TRSM-vector. With these facts, we may expect a more efficient system. We employed recall and precision commonly used in information retrieval to evaluate the effectiveness of LEX-representation. Based on our examination, we found that the effectiveness of LEX-representation is comparable with TRSM-representation while the efficiency of LEX-representation should be better than the existing TRSM-representation. We concluded that lexicon-based document representation was another alternative potentially used to represent a document while considering semantics. We are tempted to implement the LEX-representation together with linguistic computation, such as tagging and feature selection, in order to retrieve more relevant terms with high weight. With regard to the TRSM method, enhancing the quality of tolerance class is crucial based on the fact that the

[*] Address for correspondence: Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland

TRSM method is fully reliant on the tolerance classes. We plan to combine other resources such as Wikipedia Indonesia to generate a better tolerance class.

Keywords: Tolerance rough sets model, information retrieval

1. Introduction

Information retrieval (IR) is a field of study for extracting and retrieving information from a large volume of digital documents, which is usually in the form of unstructured text, based on the information need inputed by human users. The information need is represented as a query and the information retrieval system (IRS) should interpret this query and come up with a list of documents measured as relevant with regard to the information need.

It is a big challenge for an IRS to interpret the queries made by users, particularly because the common form of query consists of very few terms. Even if it comes up with more terms, there is limited information available in the first place for the IRS in order to grasp the information needed by users.

There are three main tasks basically involved in information retrieval: *a*) construct the document representation; *b*) construct the query representation; and *c*) calculate the relevance of document against query representations. Therefore, the effort of developing a semantic IRS revolves around these three tasks. Some studies [5, 7, 17] have shown that ontology makes a significant contribution to improving the representation of documents as well as the query in terms of semantic relatedness. In an interactive IRS, handling the relevance of the feedback given by the users may bring benefits to the endeavour to understand the information need [10, 11]. The relevance of documents may be measured by a similarity measure such as cosine. The other similarity measures have been listed and evaluated in Takale and Nandgaonkar's study [20].

The tolerance rough sets model (TRSM), developed by Kawasaki, Nguyen and Ho [8] is a tool to model document-term relations in text mining for a task such as information retrieval or clustering. A study of TRSM implementation to search results clustering presented by Nguyen and Ho [14] proved that the TRSM-representation created had positive effects on clustering quality. An example of a TRSM study which touched the second main task of IRS was conducted by Gaoxiang, Heling, Zhengding and Ruixuan [4] which was about query expansion. The study showed that TRSM implementation was effective and high search precision was gained. When it was first introduced, TRSM touched the first main task of IR, i.e. generating the document representation. Some studies [4, 8, 14, 23] have shown that the representation of documents yielded by TRSM (we refer to this representation as TRSM-representation) was richer than the baseline representation in terms of semantic relatedness. The baseline representation of TRSM is modelled by calculating the term frequency (TF) and the inverse document frequency (IDF) of a term, i.e. commonly called TF*IDF weighting scheme[1], hence we refer to this representation as TFIDF-representation.

TRSM employs a vector space model [12] hence it represents the document as a vector of term weight in a high dimensional space. The richer representation claimed as the benefit of TRSM means that there is less zero-valued in document vector. Despite the fact that it can increase the possibility of two documents having non-zero similarity although they do not share any terms in original document,

[1] Please see Section 2.3 for explanation of TF*IDF weighting scheme.

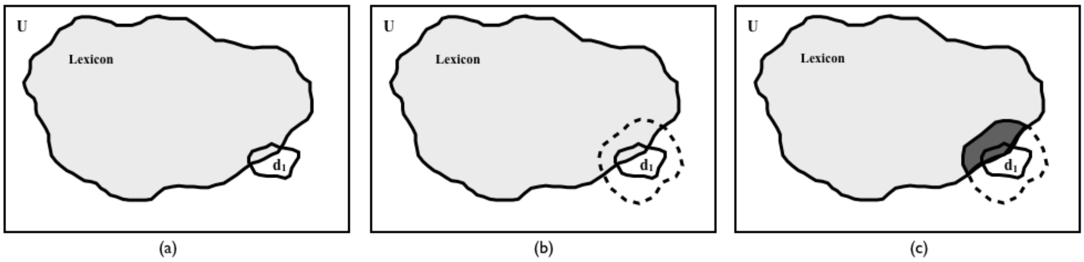


Figure 1. The idea of mapping process. Picture (a) shows relation between lexicon and a TFIDF-representation (d_1), picture (b) shows relation between lexicon and a TRSM-representation (depicted by area inside dashed line), while picture (c) shows relation between lexicon and a LEX-representation (depicted by the darkest area).

this fact leads us to a presumption that higher computational cost may become a significant trade-off in TRSM.

In our previous study [23], we showed that TRSM was able to fetch the important terms which should be retrieved by the automated process of the system. Nevertheless, based on comparison between the lexicon[2] and of the indexed terms, we identified 64.89% did not occur in lexicon; the contributors were foreign terms (mostly in English), colloquial terms (e.g. *yoi* (in formal English: yes), *terus* (in formal English: and then), *rekans* (in formal English: friends)) and proper nouns.

We propose a novel method, called a lexicon-based document representation, for a compact document representation. The heart of our method is the mapping process of terms occurring in TRSM-representation to terms occurring in lexicon, which gives us a new document representation consisting only of terms occurring in lexicon (we refer to this representation as LEX-representation) as an output. Consider Fig. 1 for depiction of the idea. Hence this method represents a document as a vector in a lower dimensional space and eliminates all informal terms previously occurring in TRSM-representation. By this fact, we can expect less computational cost. For evaluation, we take advantage of recall and precision commonly used in information retrieval to measure the effectiveness of LEX-representation. We also did manual investigation into the list of terms considered as highly related with a particular concept in order to assess the quality of the representations.

The contribution of this paper is twofold. First, we introduce a new approach to modelling document-term relations based on a lexicon which is more compact. Second, we can expect better efficiency in LEX-representation than the existing TRSM-representation.

2. Basic Theory

It is important to explain the basic idea of TRSM for better understanding of the meaning of *compact document representation* we proposed. We start our explanation with rough sets theory due to the fact that TRSM developed based on this. In order to understand the special weighting scheme used for TRSM, we gave a short explanation about the well known TFIDF term weight scheme at the end of this section.

[2] It is an Indonesian lexicon created by the University of Indonesia described in a study of Nazief and Adriani in 1996 [2] which consists of 29,337 Indonesian root words. The lexicon has been used in other studies [1, 3]

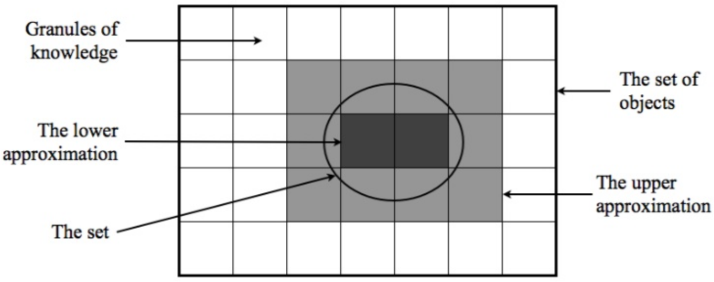


Figure 2. Basic idea of rough sets theory as it is explained in [16]

2.1. Rough Sets Theory

In 1982, Pawlak introduced a method called rough sets theory [15] as a tool for data analysis and classification. During the years, this method has been studied and implemented successfully in numerous areas of real-life applications [9]. Basically, rough sets theory is a mathematical approach to vagueness which expresses the vagueness of a concept by means of the boundary region of a set; when the boundary region is empty, it is a crisp set. Otherwise, it is a rough set [16]. The central point of rough sets theory is an idea that any concept can be approximated by its *lower* and *upper approximations*, and the vagueness of concept is defined by the region between its upper and lower approximations. Consider Fig. 2 for illustration.

Let us think of a concept as a subset X of a universe U, $X \subseteq U$, then in a given *approximation space* $A = (U, R)$ we can denote the lower approximation of concept X as $L_A(X)$ and the upper approximations of concept X as $U_A(X)$. The boundary region, $BN_A(X)$, is the difference between the upper and lower approximations, hence

$$BN_A(X) = U_A(X) - L_A(X) \tag{1}$$

Let $R \subseteq U \times U$ be an *equivalence relation* that will partition the universe into *equivalence classes*, or *granules of knowledge*, thus formal definition of lower and upper approximations are

$$L_A(X) = \bigcup_{x \in U} \{R(x) : R(x) \subseteq X\} \tag{2}$$

$$U_A(X) = \bigcup_{x \in U} \{R(x) : R(x) \cap X \neq \emptyset\} \tag{3}$$

2.2. Tolerance Rough Sets Model

Komorowski, Pawlak, Polkowski and Skowron in their paper [9] explained that the equivalence relation $R \subseteq U \times U$ of classical rough sets theory required three properties: reflexive (xRx), symmetric ($xRy \to yRx$), and transitive ($xRy \wedge yRz \to xRz$); for $\forall x, y, z \in U$, thus the universe of an object would be divided into disjoint classes. However, this requirement showed it was not always suitable for practical applications working on text, such as in information retrieval. Ho and Nguyen [6] annotated this problem clearly that association between terms was better viewed as overlapping classes (see Fig. 3), particularly when term co-occurrence was used to identify the semantic relatedness between terms.

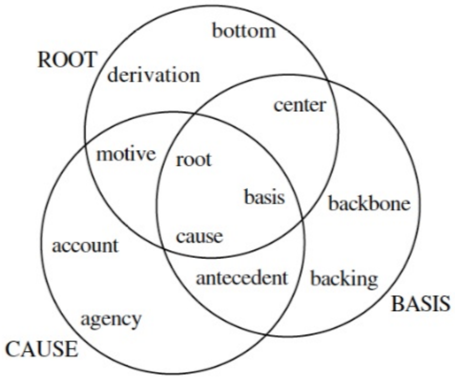


Figure 3. Overlapping classes between terms *root*, *basis*, and *cause* [6]

The overlapping classes can be generated by a relation called *tolerance relation* which was introduced by Skowron and Stepaniuk [19] as a relation in *generalized approximation space*. The generalised approximation space is denoted as a quadruple $\mathcal{A} = (U, I, \nu, P)$, where U is a non-empty universe of objects, I_θ is the uncertainty function, ν is the vague inclusion function, and P is the structurality function.

In the information retrieval context, we can assume a document 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 the tolerance relation.

In order to generate the document representation, which is claimed to be richer in terms of semantic relatedness, the TRSM needs to create tolerance classes of terms and approximations of subsets of documents. Let $D = \{d_1, d_2, ..., d_N\}$ as a set of text documents and $T = \{t_1, t_2, ..., t_M\}$ as a set of index terms from D, then the tolerance classes of terms in T created based on the co-occurrence of index terms in all documents from D. The document representation is represented as a vector of weight $d_i = \{w_{i,1}, w_{i,2}, ..., w_{i,M}\}$, where $w_{i,j}$ denotes the weight of term t_j in document d_i. The definitions of tolerance approximation space $\mathcal{R} = (T, I_\theta, \nu, P)$ are as follows

Universe: The universe U is the set of index terms T

$$U = \{t_1, t_2, ..., t_M\} = T \tag{4}$$

Tolerance class: Skowron and Stepaniuk [19] maintain that an uncertainty function $I : U \rightarrow \mathbb{P}(U)$, where $\mathbb{P}(U)$ is a power set of U, is any function from U into $\mathbb{P}(U)$ satisfying the conditions $x \in I(x)$ for $x \in U$ and $y \in I(x) \Leftrightarrow x \in I(y)$ for any $x, y \in U$. This means that we assume the relation $x\mathrm{I}y \Leftrightarrow y \in I(x)$ is a tolerance relation and $I(x)$ is a tolerance class of x.

The parameterised tolerance class I_θ is then defined as

$$I_\theta(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \theta\} \cup \{t_i\} \tag{5}$$

where θ is a positive parameter and $f_D(t_i, t_j)$ denotes the number of documents in D that contain both terms t_i and t_j. From (5), it is clear that it satisfies the condition of being reflexive ($t_i \in I_\theta(t_i)$)

and symmetric ($t_j \in I_\theta(t_i)$) required by a tolerance relation; the tolerance relation $R \subseteq T \times T$ can be defined by means of function I_θ as $t_i R t_j \Leftrightarrow t_j \in I_\theta(t_i)$. The tolerance class $I_\theta(t_i)$ represents the concept related to t_i and the precision of the concept represented is tuned by varying the threshold θ.

Vague inclusion function: the vague inclusion function $\nu : \mathbb{P}(U) \times \mathbb{P}(U) \rightarrow [0,1]$ measures the degree of inclusion between two sets, rather than the degree of membership function for objects as in fuzzy set theory. Hence, it can determine the matter whether the tolerance class $I\left(x\right)$ of an object $x \in U$ is included in a set X. The function ν must be *monotone* w.r.t the second argument, i.e. if $Y \subseteq Z$ then $\nu(X,Y) \leq \nu(X,Z)$ for $X, Y, Z \subseteq U$. The vague inclusion ν is defined as

$$\nu\left(X,Y\right) = \frac{|X \cap Y|}{|X|} \tag{6}$$

Together with the uncertainty function I, the vague inclusion function ν defines the *rough membership function* for $x \in U, X \subseteq U$ as $\mu_{I,\nu}(x,X) = \nu(I(x),X)$. Therefore, the membership function μ for $t_i \in T, X \subseteq T$ is defined as

$$\mu\left(t_i, X\right) = \nu\left(I_\theta\left(t_i\right), X\right) = \frac{|I_\theta\left(t_i\right) \cap X|}{|I_\theta\left(t_i\right)|} \tag{7}$$

Structurality function: with structurality function $P : I(U) \rightarrow \{0,1\}$, where $I(U) = \{I\left(x\right) : x \in U\}$, one can construct two subsets based on value of $P\left(I\left(x\right)\right)$, named *structural subset* and *nonstructural subset*, when $P\left(I\left(x\right)\right) = 1$ and $P\left(I\left(x\right)\right) = 0$ respectively. In TRSM, all tolerance classes of index terms are considered as structural subsets, hence for all $t_i \in T$

$$P(I_\theta(t_i)) = 1 \tag{8}$$

With the foregoing definitions, we can define the lower approximation, upper approximation, and boundary region of any subset $X \subseteq T$ in tolerance space $\mathcal{R} = (T, I_\theta, \nu, P)$ as follows

$$L_{\mathcal{R}}(X) = \{t_i \in T \mid \nu(I_\theta(t_i), X) = 1\} \tag{9}$$

$$U_{\mathcal{R}}(X) = \{t_i \in T \mid \nu(I_\theta(t_i), X) > 0\} \tag{10}$$

$$BN_{\mathcal{R}}(X) = U_{\mathcal{R}}(X) - L_{\mathcal{R}}(X) \tag{11}$$

Refers to the basic idea of rough sets theory [16], for any set of terms X, intuitively we can see the upper approximation $\mathbf{U}_{\mathcal{R}}(X)$ as the set of concepts that share some semantic meanings with X, the lower approximation $\mathbf{L}_{\mathcal{R}}(X)$ as the *core* concepts of X, while the boundary region $\mathbf{BN}_{\mathcal{R}}(X)$ consists of concepts that *cannot be classified uniquely* to the set or its complement, by employing available knowledge.

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

$$\mathbf{U}_{\mathcal{R}}(d_i) = \{t_i \in T \mid \nu(I_\theta(t_i), d_i) > 0\} \tag{12}$$

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

$$w^*_{ij} = \frac{1}{S} \begin{cases} (1 + \log f_{d_i}(t_j)) \log \frac{N}{f_D(t_j)} & \text{if } t_j \in d_i \\ 0 & \text{if } t_j \notin \mathbf{U}_{\mathcal{R}}(d_i) \\ \min_{t_k \in d_i} w_{ik} \frac{\log \frac{N}{f_D(t_j)}}{1+\log \frac{N}{f_D(t_j)}} & \text{otherwise} \end{cases} \tag{13}$$

where S is a normalisation factor. The extended weighting scheme is defined from the standard TF*IDF 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 similarities. This leads to a higher possibility of two documents having non-zero similarities although they do not share any terms. This is the main advantage the TRSM-based algorithm claims to have over traditional approaches.

2.3. TF*IDF Weighting Scheme

Salton and Buckley summarised clearly in their paper [18] the insights gained in automatic term weighting and provided baseline single-term-indexing models with which other more elaborate content analysis procedures can be compared. The main function of a term-weighting system is the enhancement of retrieval effectiveness where this result depends crucially on the choice of effective term-weighting systems. *Recall* and *Precision* are two measures normally used to assess the ability of a system to retrieve the relevant and reject the non-relevant items of a collection. Considering the trade-off between recall and precision, in practice compromises are normally made by using terms that are broad enough to achieve a reasonable recall level without at the same time producing unreasonably low precision.

Salton and Buckley further explained that, with regard to the differing recall and precision requirements, three main considerations appear important:

1. *Term frequency* (tf). The frequent terms in individual documents appear to be useful as recall-enhancing devices.

2. *Inverse document frequency* (idf). The *idf* factor varies inversely with the number of documents df_t to which a term t is assigned in a collection of *N* documents. It favors terms concentrated in a few documents of a collection and avoids the effect of high frequency terms which are widespread in the entirety of documents.

3. *Normalisation*. Normally, all relevant documents should be treated as equally important for retrieval purposes. The normalisation factor is suggested to equalise the length of the document vectors.

Table 1 summarises some of the term weighting schemes together with the mnemonic which is sometimes called SMART notation. One example of the mnemonic is *lnc.ltc*. The first triplet (i.e. *lnc*) represents the composite weight of document vector, while the second triplet (i.e. *ltc*) represents the composite weight of query vector. For each triplet, consecutively from the first to the third letter, it denotes the form of term frequency component, of document frequency component, and of normalization component. Thus, mnemonic *lnc.ltc* means that the document vector employs log-weighted term frequency, no idf for collection component, and applies cosine normalisation, while the query vector utilize a composite

Table 1. Term-weighting components with SMART notation [12]. Here, $tf_{t,d}$ is the term frequency of term t in document d, N is the size of document collection, df_t is document frequency of term t, w_i is the weight of term t in document i, u is the number of unique terms in document d, and *CharLength* is the number of characters in the document.

Component	Type	Formula
Term Frequency	n (natural)	$tf_{t,d}$
	l (logarithm)	$1 + log(tf_{t,d})$
	a (augmented)	$0.5 + \frac{0.5 \times tf_{t,d}}{max_t(tf_{t,d})}$
	b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$
	L (log ave)	$\frac{1+log(tf_{t,d})}{1+log(ave_{t \in d}(tf_{t,d}))}$
Collection Frequency	n (no)	1
	t (idf)	$log\frac{N}{df_t}$
	p (prob idf)	$max\{0, log\frac{N-df_t}{df_t}\}$
Normalization	n (none)	1
	c (cosine)	$\frac{1}{\sqrt{\sum_i (w_i)^2}}$
	u (pivoted unique)	$\frac{1}{u}$
	b (byte size)	$\frac{1}{CharLength^{\alpha}}, \alpha < 1$

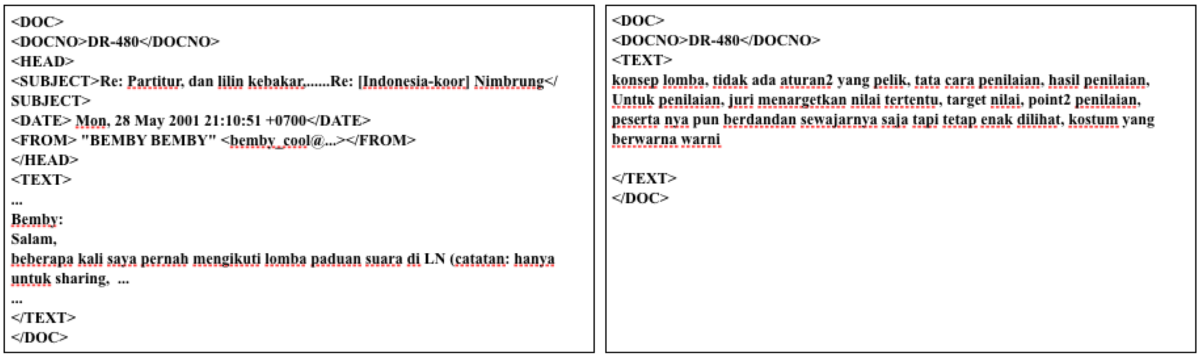
weight comprises of log-weighted term frequency, idf weighting for collection component, and cosine normalisation.

Manning et al. [12] stated that cosine similarity, i.e. (15), is fundamental to IR systems that use any form of vector space scoring. Given a query vector and a set of document vectors in a high dimensional space, we may rank the documents by comparing the angle between the query vector and each document vector; the smaller the angle, the more similar the vectors. In linear algebra, the angle θ between two vectors, $\overrightarrow{x}$ and $\overrightarrow{y}$, can be measured as follows:

$$\overrightarrow{x} \cdot \overrightarrow{y} = |\overrightarrow{x}| * |\overrightarrow{y}| * cos(\theta) \tag{14}$$

where $\overrightarrow{x} \cdot \overrightarrow{y}$ represents the *dot product* while $|\overrightarrow{x}|$ and $|\overrightarrow{y}|$ represent the lenght of the vectors. The dot product $\overrightarrow{x} \cdot \overrightarrow{y}$ of two vectors is defined as $\sum_{j=1}^{M} x_j * y_j$ and the Euclidean length of a vector $|\overrightarrow{x}|$ is defined as $\sum_{j=1}^{M} (x_j)^2$. Thus, the following formula can be used to measure the similarity between a query vector Q and a document vector D:

$$similarity(Q, D) = \frac{\sum_{j=1}^{M} w_{qj} * w_{dj}}{\sqrt{\sum_{j=1}^{M} (w_{qj})^2 * \sum_{j=1}^{M} (w_{dj})^2}} \tag{15}$$



```
<DOC>
<DOCNO>DR-480</DOCNO>
<HEAD>
<SUBJECT>Re: Partitur, dan lilin kebakar,......Re: [Indonesia-koor] Nimbrung</
SUBJECT>
<DATE> Mon, 28 May 2001 21:10:51 +0700</DATE>
<FROM> "BEMBY BEMBY" <bemby_cool@...></FROM>
</HEAD>
<TEXT>
...
Bemby:
Salam,
beberapa kali saya pernah mengikuti lomba paduan suara di LN (catatan: hanya
untuk sharing, ...
...
</TEXT>
</DOC>
```

```
<DOC>
<DOCNO>DR-480</DOCNO>
<TEXT>
konsep lomba, tidak ada aturan2 yang pelik, tata cara penilaian, hasil penilaian,
Untuk penilaian, juri menargetkan nilai tertentu, target nilai, point2 penilaian,
peserta nya pun berdandan sewajarnya saja tapi tetap enak dilihat, kostum yang
berwarna warni

</TEXT>
</DOC>
```

Figure 4. The content of corpora. On the left is an example of ICL-corpus document which consists of original document. On the right is an example of WORDS-corpus document which consists of keywords given by human expert manually for particular ICL-corpus document, i.e. in this case ICL-corpus document with number ”DR-480” which is shown on the left.

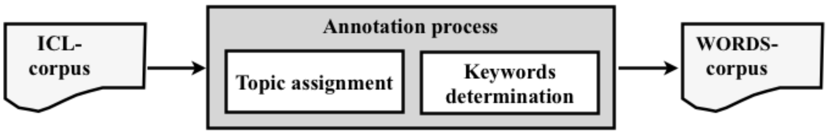


Figure 5. Corpus relationship. The WORDS-corpus was yielded by human expert in annotation process over ICL-corpus

3. Our study

The goal of our study is to investigate the performance of lexicon-based document representation. For this, we use our own corpus which is in Indonesian due to the fact that this study is part of a continuous study [22].

3.1. Corpus

We worked with two choral experts intensively in the 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. The annotation process aimed to grasp how the topic(s) could be assigned to a particular document which was mainly described by the keywords determined. We take benefit from these keywords as the list of important terms related with the topic of document (i.e. the document) and assume that the other terms not listed are less important terms. The first step of topic assignment yielded 127 topics and the keywords determination yielded a new corpus, called WORDS-corpus.

Our original corpus, called ICL-corpus, consists of 1,000 first emails of Indonesian Choral Lovers (ICL) Yahoo! Groups and are formatted as of the Text REtrieval Conference (TREC) format [24]. Therefore our test collections consist of three parts (a set of documents, a set of information needs, and rel-

Table 2. List of topics. This is a list of 28 topics in Indonesian and the total number (document frequency) of relevant documents for each topic.

ID	Topic	DF	ID	Topic	DF
0	Komenter kegiatan	80	14	Orang	16
1	Internal milis ICL	100	15	Referensi	27
2	Kompetisi	181	16	Media paduan suara	33
3	Konser	158	17	Latihan	12
4	Karya musik	125	18	Pertemuan anggota milis ICL	37
5	Perkenalan anggota milis ICL	87	19	Spam	14
6	Manajemen	46	20	Instrumen	19
7	Kelompok musik	52	21	Genre	14
8	Aplikasi	38	22	Tangga nada	18
9	Hal teknis milis	33	23	Seminar atau pelatihan	28
10	Teknik vokal	13	24	Hak cipta	11
11	Performa	14	25	Terminologi	11
12	Dokumentasi	38	26	Forum	15
13	Interpretasi karya musik	24	27	Publikasi	14

evance judgments) and all documents are marked up in a TREC-like format, i.e. *each document* is marked up by <DOC> and </DOC> tags, the *document number* is marked up by <DOCNO> and </DOCNO> tags, the *subject of email* is marked up by <SUBJECT> and </SUBJECT> tags, the *date of email* is marked up by <DATE> and </DATE> tags, the *sender* is marked up by <FROM> and </FROM> tags, and the *text body* is marked up by <TEXT> and </TEXT> tags. Consult Fig. 4 to see the content of both corpora. Notice that the main difference between documents in ICL-corpus and WORDS-corpus lies in the *text body*, i.e. the document of ICL-corpus consists of the body of email while the document of WORDS-corpus consists of keywords defined by human experts. Fig. 5 shows the relationship between both corpora.

The topic assignment yielded 127 topics of which many have low document frequency; 81.10% have document frequency less than 10 and 32.28% of them have document frequency 1. We further processed the 127-topics and came up with 28 topics as listed in Table 2. Column 1 is the *topic identifier*, column 2 is the *topic* in Indonesian, and column 3 is the *document frequency* or total number of relevant documents.

3.2. Experiment Process

There were four main phases in our study, i.e. preprocessing phase, TRSM phase, mapping phase and evaluation phase as depicted in Fig 6. Generally we did the first three phases over both corpora individually and analysed them in the evaluation process.

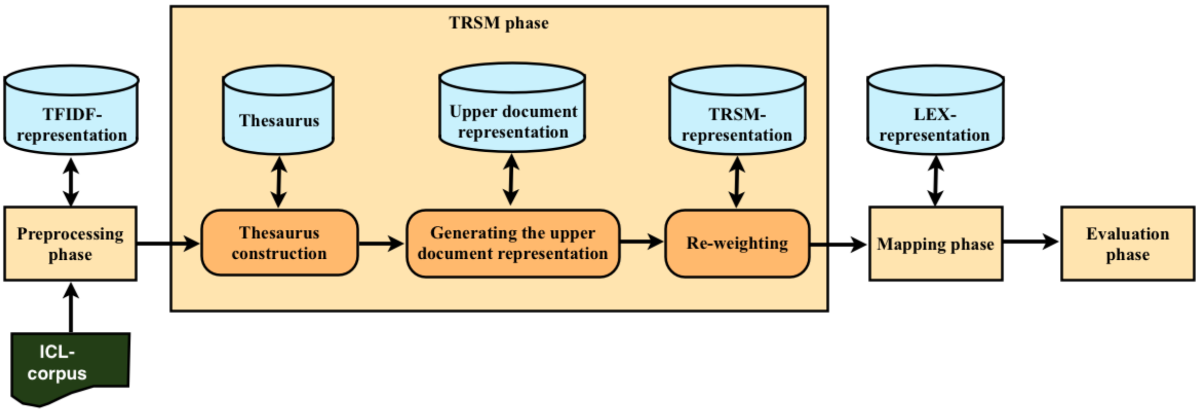


Figure 6. Main process in the study. The process consisted of 4 phases: preprocessing phase, TRSM phase, mapping phase, and evaluation phase.

3.2.1. Preprocessing Phase

The goal of this phase is to generate document representation based on the TF*IDF weighting scheme. This is a phase when we did tokenisation, stemming, stopword elimination, and finally generated TFIDF-representation. This phase was powered by Lucene[3], in the sense that we put the stopword list and stemmer into it and let Lucene do the job.

We chose to implement Vega's stopword list [21] as it showed a positive contribution to an Asian study [3], i.e. the use of Vega's stopword list gave the highest precision and recall among other Indonesian stopword lists. We used the so called Confix-Stripping Stemmer (CS-Stemmer), a version of Indonesian stemmer, which has been shown by Adriani et al. [1] to be the most accurate stemmer among other automated Indonesian stemmers. In order to work, the CS-Stemmer requires a dictionary called DICT-UI. The study showed that the use of DICT-UI tended to produce more accurate results than the use of the other dictionary, i.e. *Kamus Besar Bahasa Indonesia* (KBBI)[4]. The DICT-UI is actually the lexicon of this study.

3.2.2. 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 the tolerance matrix comprises tolerance classes of all terms based on (5). For the purpose of this study, we altered the tolerance value θ from 0 to 50.

2. Create the upper approximation of documents $U_{\mathcal{R}}(d_i)$ using (12).

[3] Lucene is an information retrieval library, which is free, open source project implemented in Java, and a project in the Apache Software Foundation, licensed under the liberal Apache Software License [13]. For this study, we used Lucene.3.1.0, downloaded from http://lucene.apache.org/core/downloads.html.

[4] KBBI is a dictionary copyrighted by *Pusat Bahasa* (in English: Language Center), Indonesian Ministry of Education, which consists of 27,828 root words

3. Generate the TRSM-representations by recalculating the TFIDF-representations using (13) and considering the upper approximation of documents $U_{\mathcal{R}}(d_i)$.

Let us call the TRSM-representation for ICL-corpus and WORDS-corpus as *ICL-TRSM-representation* and *WORDS-TRSM-representation* respectively.

3.2.3. Mapping Phase

Our intention in this phase is to map the index terms of TRSM-representation into the terms of the lexicon.

We noticed that the total number of terms in the lexicon (29,337 terms) was much bigger than the total number of index terms in ICL-corpus (9,742 terms) and WORDS-corpus (3,477 terms). We also noted that tolerance class was calculated based on co-occurrence of terms between documents in a corpus, hence there would be no relationship to other terms outside the corpus. Further more, there must be an intersection between lexicon and each document in a corpus because all documents must have some *formal* terms in order to be understood. Consequently, there would be no benefit in considering all terms in the lexicon during the mapping process.

In order to make the process faster, we intersected the lexicon with each corpus and called the result as *known-terms* K. Let $B = \{b_1, b_2, ..., b_P\}$ is a set of terms in the lexicon, P is the total number of terms in the lexicon, and C is the total number of terms in known-terms, then $K = \{t_i \in T \mid t_i \cap b_j\} = \{k_1, k_2, ..., k_C\}$, for all $b_j \in B$. The terms which occurred in known-terms then became the index terms in LEX-representation. The total number of known-terms for ICL-corpus and WORDS-corpus were 3,444 and 1,566 respectively.

Let N denote total number of documents in corpus and M denote total number of terms in corpus, then the mapping process was conducted as follows

Input: matrix of TRSM-representation $TRSM_{matrix} = [trsm_{i,j}]_{NxM}$ for all $t_j \in T$ and $d_i \in D$, where $trsm_{i,j}$ denotes weight of term t_j in document d_i.

Output: matrix of LEX-representation $LEX_{matrix} = [lex_{i,l}]_{NxC}$ for all $k_l \in K$ and $d_i \in D$, where $lex_{i,l}$ denotes weight of term k_l in document d_i.

Process: generate LEX_{matrix} based on (16) for all $t_j \in T$, $k_l \in K$, and $d_i \in D$

$$lex_{i,l} = \begin{cases} trsm_{i,j} & \text{if } k_l = t_j \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

We should mention that, during the annotation process, we found that our human experts seemed to encounter difficulty in determining keywords. Thus, rather than listing the keywords, our experts chose sentence(s) from the text or made their own sentence(s). This action made the WORDS-corpus contain both formal and informal terms. Based on this fact, we decided to run the mapping process not only on ICL-corpus but also on WORDS-corpus, in order to remove the informal terms occurring in both corpora. Let us call the resulting representation *ICL-LEX-representation* and *WORDS-LEX-representation*.

3.2.4. Evaluation Phase

There were two tasks committed in the evaluation phase. We named them categorisation and calculation. Categorisation was the task when we clustered documents of the same topics together. The motivation behind this task was based on the annotation process conducted by our human experts, i.e. keywords determination for each document. Thus, we perceived each topic as a concept and considered the keywords in WORDS-corpus as variants of terms semantically related with a particular concept. For this task, we used the 127-topics defined by our human experts, therefore we got 127 classes. Let us call the output of this process *ICL-topic-representation* and *WORDS-topic-representation* for each corpus. Technically, those representations were topic-term matrices.

Manning, Raghavan, and Schütze in their book [12] stated that recall and precision are the most frequent and basic measures for information retrieval effectiveness. They defined recall as the fraction of relevant documents that are retrieved while precision is the fraction of retrieved documents that are relevant.

$$Recall = \frac{\sharp(relevant\ items\ retrieved)}{\sharp(relevant\ items)} \tag{17}$$

$$Precision = \frac{\sharp(relevant\ items\ retrieved)}{\sharp(retrieved\ items)} \tag{18}$$

In a calculation task, we used the notions of recall and precision in terms of calculating the *documents* as well as the *terms*. The first calculation computed the *terms* while the second calculation computed the *documents*. Thus, for the *terms-calculation*, the recall *R* is the fraction of relevant *terms* that are retrieved while precision P is the fraction of retrieved *terms* that are relevant. Notice that our WORDS-corpus consists of keywords defined by human experts, hence we considered WORDS-corpus as the *ground truth*, i.e. WORDS-corpus consists of *relevant terms* which should be retrieved by automated system.

Briefly, in the *terms-calculation*, we categorised LEX-representation of both corpora and then computed the recall and precision of topic-representations generated with and without a mapping process. Whereas in the *documents-calculation*, we computed the standard recall and mean average precision (MAP) for all representations.

4. Result

4.1. Calculating the Terms

Based on the *terms-calculation*, our findings are summarised by Fig. 7. Those graphs show that the mean of recall and precision values across 127 topics vary by the alternation of tolerance values θ and tend to be smaller as the tolerance value becomes higher.

We have mentioned in section 3.2.4 that in the *terms-calculation* we focused on *terms* rather than *documents* when calculating recall and precision. Instead of document representation, the recall and precision values were computed over the terms of topic-representation. We measured the quality of topic-representation of ICL-corpus based on the occurrence of relevant terms in it; the relevant terms were the index terms of topic-representation of WORDS-corpus.

Pertaining to the mapping process, we perceive the recall as a value which expresses the ability of the mapping process to keep the relevant terms out of the irrelevant ones. Thus, from Fig. 7(a) we can

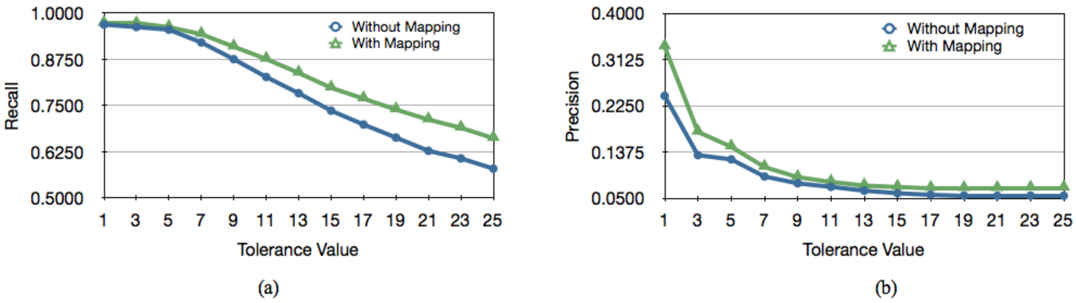


Figure 7. Mean of Recall and Precision. Graph (a) shows the mean of recall values, while graph (b) shows the mean of precision values. The mean were calculated over 127 topics.

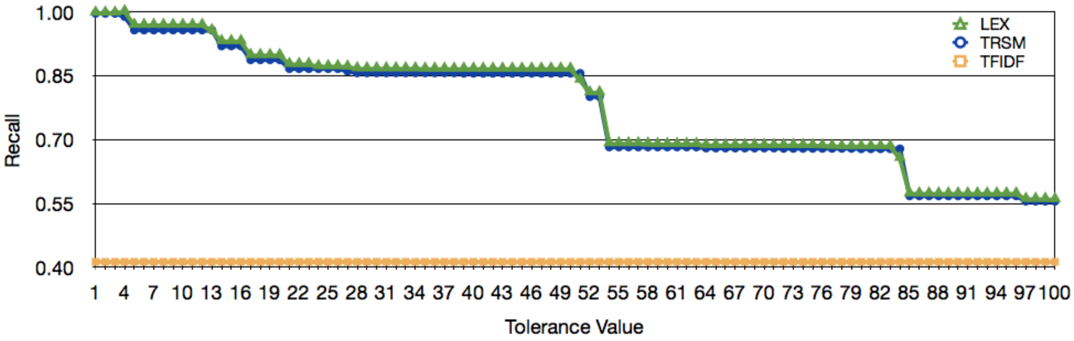


Figure 8. Recall. This graph shows the recall values based on TFID-representation, TRSM-representation, and LEX-representation.

say that the mapping process outperforms the original TRSM method in terms of preserving the relevant terms. A gradual reduction of the ability is shown as the tolerance value θ gets higher, yet the mapping process seems to work better.

From another point of view, by the nature of TRSM method, a greater tolerance value should increase the number of index terms discarded from being introduced into the document representation. Considering Fig. 7(b), the behavior seems to shield not only the irrelevant index terms but also the relevant ones to be chosen to extend the base representation, although at some point the change is not significant anymore, which happens at $\theta > 17$. However, the mapping process performs better once again in this figure.

4.2. Calculating the Documents

In this task, the standard recall and precision were computed using the *trec_eval*[5] program based on TFIDF-representation, TRSM-representation and LEX-representation of ICL-corpus over 28 topics for $\theta = 1$ to 100. Figure 8 is the graph of recall while Fig. 9 is the graph of mean average precision (MAP). In the figures, `LEX` is the LEX-representation, `TRSM` is the TRSM-representation, and `TFIDF` is

[5] It is publicly available on `trec.nist.gov/trec_eval`. We used the `trec_eval.9.0`.

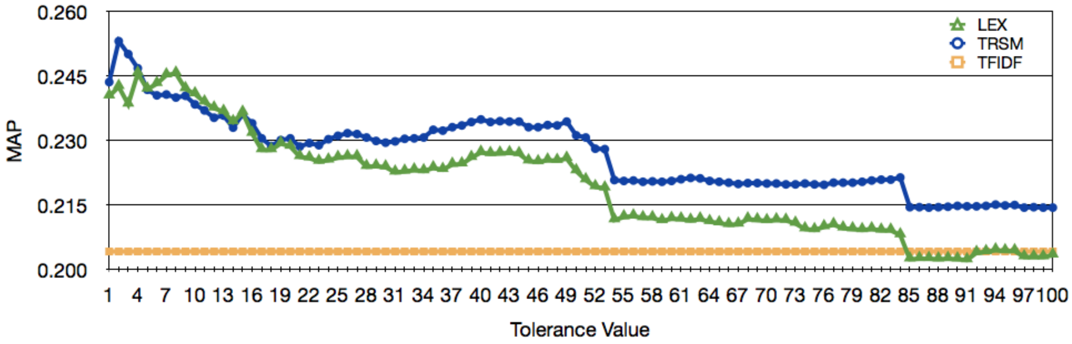


Figure 9. Mean Average Precision. This graph shows the mean average precision (MAP) values based on TFID-representation, TRSM-representation, and LEX-representation.

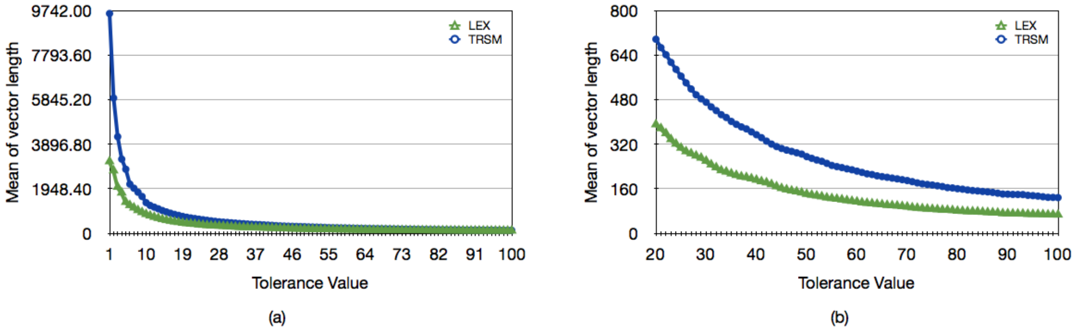


Figure 10. Length of vector. Graph (a) shows the mean length of TRSM-representation and LEX-representation for $\theta = 1$ to 100 while graph (b) is the inset of graph (a) for $\theta = 20$ to 100.

the TFIDF-representation.

Figure 8 displays that LEX-representation works better than TFIDF-representation, even has slightly higher recall values than TRSM-representation at almost all level of $\theta = 1$ to 100. The trade-off to the recall values can be seen in Fig. 9. Here, the performance of LEX-representation is shown to be similar with TRSM-representation on low tolerance values ($\theta < 22$) and has slightly better precision at $\theta = 5$ to 15. Compared with TFIDF-representation, it performs better at $\theta < 85$.

The result depicted by Fig. 8 and Fig. 9 looks consistent with the result presented by Fig. 7. Those figures say that the increment of tolerance value leads to the less relevant terms in topic-representation and the more incapable the system to retrieve relevant documents. Even though the mapping process proved to be more capable to maintain the relevant terms and the recall value of LEX-representation have proportional result with of TRSM-representation, the mean average precision (MAP) value shows that TRSM-representation performs better in general. Figure 10, which presents the mean length of TRSM-representation and LEX-representation for tolerance values between 1 and 100, seems to explain that the vector length has contribution at some degree.

Figure 10 tells us that document representation of TRSM tends to be longer than document representation yielded by mapping process. In fact, our observation through all document vectors for $\theta = 1$

to 100 yielded $\overrightarrow{x} \geq \overrightarrow{y}$, where $\overrightarrow{x}$ is document vector of TRSM and $\overrightarrow{y}$ is document vector of mapping process. It is not a surprising result due to the fact that mapping process conducted based on TRSM, in which the index terms of LEX-representation are those of TRSM-representation which appear in the lexicon.

The document ranking method we used in this study is the Cosine similarity method (15), which implies that the largest value of $similarity(Q, D)$ are obtained when the query Q and the document D are the same. Refer to this method, longer vector should have more benefit than the shorter one. Therefore, it is predictable that TRSM-representation outperforms the others when document vector of TRSM is the longest.

It is interesting though that at some levels between tolerance values 1 to 100 the LEX-representation has better performance than of TRSM. So, instead of the vector length, there must be another factor which give significant contribution to similarity computation based on Cosine method. The investigation went further to the tolerance classes which constructed the thesaurus.

4.3. Tolerance Class

We picked 3 topics out of 28 which were the most frequent topics in ICL-corpus as it appears in Table 2. These were *kompetisi* (in English: competition), *konser* (in English: concert), and *karya musik* (in English: musical work), and made an assumption that those topics were concepts which could be represented by a single term for each, namely *kompetisi*, *konser*, and *partitur* (in English: musical score)[6].

We generated the tolerance classes of those terms at several particular tolerance values, i.e. θ = 2, θ = 8, θ = 41, and θ = 88[7]. Specifically, we generated all terms considered semantically related with terms *kompetisi*, *konser*, and *partitur* (based on its occurrence in thesaurus) which appeared on the most relevant document retrieved by the system for each particular topic (i.e. *kompetisi*, *konser*, and *karya musik* respectively). Let us call this term sets as `TolClass_in_document`.

Table 3 and Table 4 summarise the results; column 1 lists the terms being investigated, while `TFIDF`, `TRSM`, and `LEX` columns present the number of related terms appeared in TFIDF-representation, TRSM-representation, and LEX-representation sequentially (i.e. the cardinality of `TolClass_in_document`). When $\theta = 2$ we considered those representations with regard to the top-retrieved document calculated based on TRSM model. In similar fashion, for $\theta = 8$, $\theta = 41$, and $\theta = 88$, we considered ones with regard to the top retrieved document based on mapping process, TRSM method, and base model[8]. The `Total` column is the cardinality of particular tolerance class in thesaurus. In other words, it specifies the total terms defined semantically related with term *kompetisi*, *konser*, and *partitur* at $\theta = 8$, $\theta = 41$, and $\theta = 88$.

In a glance we should notice that document vector of TRSM consists of most related terms defined in thesaurus, even at high tolerance value ($\theta = 88$) it includes all of them. It is also clear in both tables that the cardinality of `TolClass_in_document` in TFIDF-representation (showed by the `TFIDF` columns) are mostly the least.

[6] The index terms of thesaurus are in the form of single term, hence we choose term *partitur* as the representative of the *karya musik* concept.

[7] Figure 9 serves as a basis for the choice of θ values in which the TRSM-representation, LEX-representation, TRSM-representation, and TFIDF-representation outperform the other representations at θ = 2, θ = 8, θ = 41, and θ = 88 in respective order. However, particularly at θ = 88, the TFIDF-representation only performs better than the LEX-representation.

[8] The base model means that we employed the TF*IDF weighting scheme without TRSM implementation nor the mapping process.

Table 3. Total number of terms considered as highly related with terms *kompetisi*, *konser*, and *partitur* at tolerance values 2 and 8 in a top-retrieved document representation generated based on TF*IDF weighting scheme (`TFIDF`), TRSM model (`TRSM`) and mapping process (`LEX`). The `Total` column is the total terms of respective tolerance class in thesaurus.

	$\theta = 2$				$\theta = 8$			
Term	TFIDF	TRSM	LEX	Total	TFIDF	TRSM	LEX	Total
Kompetisi	54	1,587	883	1,589	31	315	203	320
Konser	37	3,508	1,664	3,513	23	902	513	909
Partitur	141	2,023	1,037	2,030	30	590	325	597

Table 4. The number of terms considered as highly related with terms *kompetisi*, *konser*, and *partitur* at tolerance values 41 and 88 in a top-retrieved document representation generated based on TF*IDF weighting scheme (`TFIDF`), TRSM model (`TRSM`) and mapping process (`LEX`). The `Total` column is the total terms of respective tolerance class in thesaurus.

	$\theta = 41$				$\theta = 88$			
Term	TFIDF	TRSM	LEX	Total	TFIDF	TRSM	LEX	Total
Kompetisi	4	7	4	7	1	1	1	1
Konser	4	92	46	96	3	21	7	21
Partitur	18	54	23	56	1	4	2	4

In order to assess the quality of document vector in terms of the relevant terms, we manually made a short list of terms we considered as semantically related with terms *kompetisi*, *konser*, and *partitur*. Table 5 displays the lists. By cross referencing our manual list with the `TolClass_in_document`, we found that `TolClass_in_document` consists of at least one term of our manual list. And as predicted, the `TolClass_in_document` of TRSM includes our terms the most.

Let us focus on Table 3 when tolerance value is 8. At $\theta = 8$, refers to Fig. 9, the LEX-representation performs better than the others, whereas refers to Fig. 10 the mean length of LEX-representation vectors is shorter than of TRSM. Note that the cardinality of `TolClass_in_document` of mapping process (showed by the `LEX` column in Table 3 and Table 4) for those three terms are smaller than of TRSM. A close observation to the vectors as well as the `TolClass_in_document` turned out that the length difference of both vectors was not too big and most of our manual terms (listed in Table 5) were found to sit on top ranks.

Indeed, based on the nature of mapping process, all of relevant terms we confronted occurred in `TolClass_in_document` of mapping process were always at higher rank than of TRSM. It happened because the index terms of LEX-representation were actually those of TRSM-representation which were not dropped out by the lexicon's.

Further, manual inspection yielded that numerous terms in `TolClass_in_document` of TRSM were remotely related to the terms *kompetisi*, *konser*, and *partitur*. With regard to the problem we mentioned in the beginning of this chapter (i.e. the existence of informal terms such as foreign terms, colloquial terms, and proper nouns), the LEX-representation had the most satisfactory result, i.e. it contained only the formal terms, which were index terms of lexicon.

Table 5. The list of index terms considered manually as highly related with terms *kompetisi*, *konser*, and *partitur*. The last column is the comparable English translation for each related index term mentioned in the middle column.

Term	Related index terms	Comparable English translation (in respective order)
Kompetisi	*kompetisi, festival, lomba, kategori, seleksi, juri, menang, juara, hasil, atur, nilai, jadwal, serta*	competition, festival, contest, category, selection, jury, win, champion, result, regulate, grade, schedule, participate
Konser	*konser, tiket, tonton, tampil, informasi, kontak, tempat, publikasi, poster, kritik, acara, panitia*	concert, ticket, watch, perform, information, contact, place, publication, poster, criticism, event, committee
Partitur	*partitur, lagu, karya, musik, koleksi, aransemen, interpretasi, komposisi, komposer*	musical score, song, creation, music, collection, arrangement, interpretation, composition, composer

We may infer now, when the total terms in LEX-representation is not in big difference with the total terms in TRSM-representation, we might expect better performance from LEX-representation, which has shorter length but the same relevant terms whose ranks are higher, or in other words, which is more compact. It is practically feasible to improve the quality of LEX-representation by processing the terms more carefully in the preprocessing phase which was left untouched in this study.

4.4. Time and Space Complexity

The computation cost of constructing the tolerance classes is $O(NM^2)$ [14]. In order to generate the LEX-representation, we need to construct the upper document representation and the TRSM-representation which are both $O(NM)$. Going from TRSM-representation to LEX-representation the computation cost is also $O(NM)$. After all, the total cost of mapping process is $O(NM^2)$.

We have mentioned before that the total number of index terms in ICL-corpus was 9,742 and WORDS-corpus was 3,477. As a result, the total number of index terms of TRSM-representations for ICL-corpus and WORDS-corpus were the same, 9,742 and 3,477 respectively. After the mapping process, we found that the total number of index terms in both corpora were reduced significantly, 64.65% for ICL-corpus and 54.93% for WORDS-corpus. The mapping process reduces the dimensionality of document vector quantitatively, thus we might expect more efficient computation when we further process the LEX-representation, e.g. for retrieval, categorization, or clustering process. The use of LEX-representation should give much benefit in applications when efficiency is put on the high priority.

5. Conclusion and Future Study

We have presented a novel approach for an alternative to a compact document representation by employing the TRSM method and then run the mapping process. The mapping process is the process of mapping the index terms in TRSM-representation to terms in the lexicon.

We evaluated the LEX-representation based on the terms of topic-representation as well as of document representation. By a comparison between topic-representation with and without mapping we have seen that the mapping process produced a better representation of document, concerning its nature ability to preserve the relevant terms. The use of LEX-representation should lead to an effective process of retrieval due to the fact that the mean of recall and precision calculation gave comparable results with TRSM-representation. We might also expect a more efficient process of retrieval based on the finding that LEX-representation has much lower dimensional space than TRSM-representation.

We conclude that the result of this study is promising. The fact that we did not implement any other linguistic computation arose our confident that those activities (such as tagging, feature selection, n-gram) might give us benefit in the effort of refining the thesaurus which serves as the basis of tolerance rough sets model, and thus in the framework of IRS, the knowledge of the system. We plan to combine other resources such as Wikipedia Indonesia to generate better tolerance classes.

Acknowledgement.

This work was supported by the Polish National Science Centre grant 2011/01/B/ST6/03867 and by the Polish National Centre for Research and Development (NCBiR) under SYNAT - Grant No. SP/I/1/77065/10 in the frame of the strategic scientific research and experimental development programme: “Interdisciplinary System for Interactive Scientific and Scientific-Technical Information”. It was also supported by Duta Wacana Christian University, Indonesia, contract No. 029/C.10/UKDW/2012. We thank the Faculty of Computer Science, University of Indonesia, for their permission to use the CS stemmer.

References

[1] Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M. M., Williams, H. E.: Stemming Indonesian: A confix-stripping approach, *ACM Transactions on Asian Language Information Processing*, **6**, December 2007, 1–33, ISSN 1530-0226.

[2] Adriani, M., Nazief, B.: Confix-Stripping: Approach to Stemming Algorithm for Bahasa Indonesia, 1996, Internal publication.

[3] Asian, J.: *Effective Techniques for Indonesian Text Retrieval*, Ph.D. Thesis, School of Computer Science and Information Technology, RMIT University, March 2007, Doctor of Philosophy Thesis.

[4] 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), 2006, 1167–1171.

[5] Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E. G. M., Milios, E.: Information Retrieval by Semantic Similarity, *Intern. Journal on Semantic Web and Information Systems (IJSWIS)*, **3**(3), 2006, 55–73.

[6] Ho, T. B., Nguyen, N. B.: Nonhierarchical Document Clustering Based on a Tolerance Rough Set Model, *International Journal of Intelligent Systems*, **17**(2), February 2002, 199–212.

[7] Janusz, A., Świeboda, W., Krasuski, A., Nguyen, H. S.: Interactive Document Indexing Method Based on Explicit Semantic Analysis, in: *Rough Sets and Current Trends in Computing* (J. Yao, Y. Yang, R. Słowiński, S. Greco, H. Li, S. Mitra, L. Polkowski, Eds.), vol. 7413 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012, ISBN 978-3-642-32114-6, 156–165.

[8] Kawasaki, S., Nguyen, N. B., Ho, T. B.: Hierarchical Document Clustering Based on Tolerance Rough Set Model, *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD ’00, Springer-Verlag, London, UK, 2000, ISBN 3-540-41066-X, 458–463.

[9] Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: *Rough Sets: A Tutorial*, Springer-Verlag, 1998, 3–98.

[10] Lv, Y., Zhai, C.: Adaptive relevance feedback in information retrieval, *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, ACM, New York, NY, USA, 2009, ISBN 978-1-60558-512-3, 255–264.

[11] Lv, Y., Zhai, C.: Positional relevance model for pseudo-relevance feedback, *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, ACM, New York, NY, USA, 2010, ISBN 978-1-4503-0153-4, 579–586.

[12] Manning, C. D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*, Cambridge University Press, 2008, ISBN 9780521865715.

[13] McCandless, M., Hatcher, E., Gospodnetić, O.: *Lucene in Action*, Manning Publications Co., 2010, ISBN 9781933988177.

[14] Nguyen, H. S., Ho, T. B.: *Rough Document Clustering and the Internet*, chapter 47, John Wiley & Sons Ltd., 2008, 987–1003.

[15] Pawlak, Z.: Rough Sets, *International Journal of Computer and Information Science*, **11**(5), 1982, 341–356.

[16] Pawlak, Z.: Some Issues on Rough Sets, *Transactions on Rough Sets I* (J. F. Peters, A. Skowron, J. W. Grzymala-Busse, B. Kostek, R. W. Swiniarski, M. S. Szczuka, Eds.), 3100, Springer, 2004, ISBN 3-540-22374-6, 1–58.

[17] Paz-Trillo, C., Wassermann, R., Braga, P. P.: An Information Retrieval Application using Ontologies, *Journal of the Brazilian Computer Society*, **11**(2), 2005, 17–31.

[18] Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval, **24**(5), Aug 1988, 513–523, ISSN 0306-4573.

[19] Skowron, A., Stepaniuk, J.: Tolerance Approximation Spaces, *Fundam. Inf.*, **27**, August 1996, 245–253, ISSN 0169-2968.

[20] Takale, S. A., Nandgaonkar, S. S.: Measuring semantic similarity between words using web search engines, *Proceedings of the 16th international conference on World Wide Web*, WWW '07, ACM, New York, NY, USA, 2007, ISBN 978-1-59593-654-7, 757–766.

[21] Vega, V. B.: *Information Retrieval for the Indonesian Language*, Master Thesis, National University of Singapore, 2001, Unpublished.

[22] Virginia, G., Nguyen, H. S.: Automatic Ontology Constructor for Indonesian Language, *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '10, IEEE Computer Society, Washington, DC, USA, 2010, ISBN 978-0-7695-4191-4, 440–443.

[23] Virginia, G., Nguyen, H. S.: Investigating the Effectiveness of Thesaurus Generated Using Tolerance Rough Set Model, *Proceedings of the 19th International Conference on Foundations of intelligent systems*, ISMIS'11, Springer-Verlag, Berlin, Heidelberg, 2011, ISBN 978-3-642-21915-3, 705–714.

[24] Voorhees, E. M., Harman, D.: Overview of the Ninth Text REtrieval Conference (TREC-9), *Proceedings of the Ninth Text REtrieval Conference (TREC-9*, National Institute of Standards and Technology (NIST), 2000, 1–14.