Document Searching Engine Using Term Similarity Vector Space Model on English and Indonesian Document

Andreas Handojo, Adi Wibowo, Yovita Ria

Informatics Engineering Department
Faculty of Industrial Technology, Petra Christian University
Jl. Siwalankerto 121-131, Surabaya 60236, Indonesia
handojo@petra.ac.id, adiw@petra.ac.id

Abstact. In line with technology development, the number of digital documents increase significantly, this will make process to the search a particular documents experience a little problem. Therefore, the role of search engines is become inevitable.
Usually, search engines conduct a searching process simply by looking at the similarities between keywords (that inputed by user) and terms in a document. In this research, we try to implement Term Similarity Vector Space Model (TSVSM), a method that also saw the relationship between the terms in the document. The relationship between terms in a document is calculated based on the frequency of occurrence in a paragraph. So this will make possible to search documents that do not contain the exact keywords that inputed, but have terms that related to those keywords.
We also try to implement TSVSM to English language documents from CiteseerX journal collection [1]. In this research we also want try it to Indonesian language documents from journal collection on Petra Christian University Research Center (both in pdf format, with total 14.000 documents). This application was built using Microsoft Visual Basic.Net 2005 and PHP.
Based on testing, this method can establish relationships between related terms that can find documents that do not contain keywords but contains terms that relate to the keyword. Time that needed to search document in Indonesian language journal is relative longer than in English language journal.

Keywords: Term Similarity Vector Space Model, Term Co-Occurrence Vector, Term Co-Occurrence Matrix, Search Engine.

1 Introduction

With the development of technology, the data storage in physical form began to change into digital storage to simplify storage and retrieval at the future. To search a digital document that needed, we usually using a search engine that is generally

looking terms in a document that have the same terms to the keyword that entered by the user (user queries). Because the search engines generally do not looking the possibility of similarity in terms in the document.
For example, when a user searches for documents with the keyword "computer" then it will only display documents that have the keyword "computer". But there still are some documents that do not have the keyword "computer" but associated with the keyword "computer" such as "laptop", "notebook", "software", etc. This resulted will reduce recall in searching the relevant documents that given by keywords.
To overcome these problems, in this research we will try to use Term Similarity Vector Space Model (TSVSM) which have assumed that each term has a relationship to another term so that the search process is not only based on the keywords that are inputed by user, but also search the relationships between terms in the documents that stored the database. Relation or similarity between the terms is calculated based on the frequency of a term appears simultaneously with the other terms in a paragraph.
The results of these search engines will be display as a list of documents that relevant to the keyword, in order of its relevance. Through this method the recall of search results expected to be improved. This study is a continuation of our previous studies about search engine using Topic-Based Vector Space Model [2].
The parsing process will be done by reading the document (in pdf format) using PDFBox API library [3]. Then the stemming process will be done using Porter Stemming algorithm [4]. For the repository we used Oracle 10g database, while applications created using Microsoft Visual Basic .Net 2005 (application parsing and indexing) and Microsoft ASP .Net 2005 (application engines).
Like classical Vector Space Model method, the Term Similarity Vector Space Model also assumes that each document is represented a number of dimensions corresponding vector with the number of terms in the document. The difference is the assumption in the calculation of the relationship between terms, where a term is assumed to be a vector whose dimensions established by the terms of the other on the vector space [5]. So the dimensions of the term vector can be obtained through the creation of co-occurrence vector.

2 VECTOR SPACE MODEL

Vector Space Model is an algorithm that can be used to measure the similarity of documents with keywords or other documents which are represented in the form of vectors [6]. This model is based on the idea that the meaning of a document formed by the words that contained. One of the improvements on the vector space model is to expand the document vector and keyword vector. In the Vector Space Model, documents and keywords can be defined as follows [7]:

$$d_j = \left(w_{1,j}, w_{2,j}, w_{3,j}, \ldots, w_{t,j}\right) \quad (1)$$

$$q = \left(w_{1,q}, w_{2,q}, w_{3,q}, \ldots, w_{t,q}\right) \quad (2)$$

Each dimension that created document vector or keyword vector is determined by the existing of the terms. Usually the term is assumed as a single word, keyword, or

phrase. If a word is considered as a term, then the number of dimensions from a doc-document is the number of words that exist in the document. Several ways to calculate the weight of each dimension is known as a term weighting that always have been developed. One way of term weighting is Term Frequency-Inverse Document Frequency (TF-IDF).

3 TERM SIMILARITY VECTOR SPACE MODEL

Similarity Term Vector Space Model is the development of the Vector Space Model which assumes that the terms on a document is also a vector which the dimensions established by the other terms contained in the vector space [6]. A term co-occurrence matrix is calculated directly from the document to establish basic based to calculating the term similarity [7]. The calculation of term co-occurrence matrix is seen from the frequency of occurrence of a term with another term in each paragraph. As shown in figure 1, where [7]:

1. Each row in this matrix represents the distribution of a term x_i.
2. Each column represents another distribution, y_j term that appears near term x_i.
3. Interchanges rows i and columns j, m_{ij} is the co-occurrence frequency between x_i and y_j is extracted from the document collection.

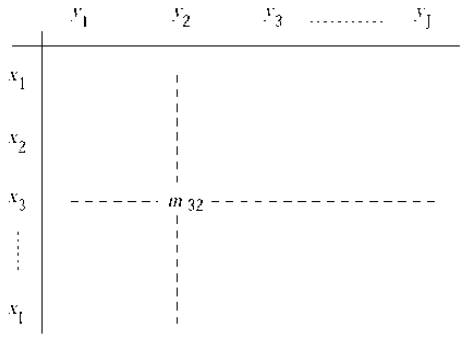


Fig. 1. Co-occurrence Matrix

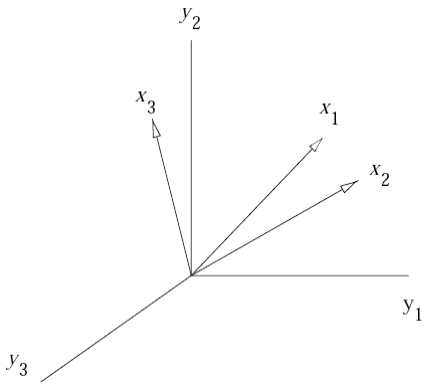


Fig. 2. Co-Occurrence Matrix Representation in Vector

Each dimension of this vector is associated with the word yj that represent the column matrix. In figure 2 is described as third-term representation of the first vector x1, x2, and x3 in figure 1 in a dimension that associated with the words y1, y2, and y3 from the collection of documents [7].

Similarity between terms can be seen from the large of cosine angle which is owned by the two term vectors that be compared. If the two vectors that term have a high similarity, the two terms are considered to have a relationship [7]. The formula for similarity searching process as follows:

$$sim(\vec{a}, \vec{b}) = \cos q = \frac{\sum_{i=1}^{N} W_{i,a} x W_{i,b}}{\sqrt{\sum_{1=1}^{N} W^2{}_{i,a}} x \sqrt{\sum_{1=1}^{N} W^2{}_{i,b}}} \quad (3)$$

Description:

$\vec{a}$: Vector term a

$\vec{b}$: Vector term b

$W_{i,a}$: Dimension of the i term in the vector-a

$W_{i,b}$: Dimension of the i term in the vector-b

N : Number dimensions of the vector term

Similarity searches have been done after finding the term vector of the co-occurrence matrix. After the words, documents, and keywords are represented as vectors in the same vector space, the keyword vectors will be compared with each vector of the document. The closest document vector to the vector keyword in the space will be delivered to users as an answer. The closeness of each vector viewed from the angle cosine between the documents with the formula as follows:

$$sim(dj, q) = \cos q = \frac{\sum_{i=1}^{N} W_{i,j} x W_{i,q}}{\sqrt{\sum_{1=1}^{N} W^2{}_{i,j}} x \sqrt{\sum_{1=1}^{N} W^2{}_{i,q}}} \quad (4)$$

Description:

dj : vector of the document j

q : vector of keyword

wi,j : i dimension of document vectors j

wi,q : i dimension of the keyword vector

4 Precision and Recall

Precision and recall is one of the methods that used to measure the performance from an information retrieval system. These measurements will compare between the relevant documents that obtained from the search results and the total number of documents [8]. This precision will be calculation using the following formula:

$$\text{Precision} = \frac{relevant\, document\, retrieved}{document\, retrieved} \quad (5)$$

Recall is number of relevant documents that obtained from search results divided by the total number of relevant document contained in the entire collection of documents.

$$\text{Re}\,call = \frac{relevant\,document\,retrieved}{total\,relevant\,document} \quad (6)$$

Ideally, recall and precision should be worth one, mean that the system gives the results of all relevant documents without showing any irrelevant. Unfortunately, in reality this is almost impossible to achieve.

5 System Design and Implementation

Design system flowchart from Document Searching Engine application can be shown on Figure 3.

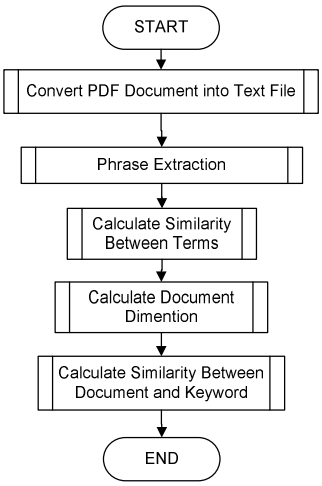


Fig. 3. Flowchart Searching Engine System

Description:
1. Convert PDF Document into Text File
In this process PDFBox library is used to convert pdf files into text files. This library is a library from open source Java programming language. This library will be running on the programming language that used for this search engine (Microsoft Visual Studio .Net 2005 and IKVM .Net libraries [9]).
2. Phrase Extraction
After changing pdf documents into text, then the next process is phrase extraction. This process is to find a set of words that have more meaning that can be seen from the frequency of collection of these words that appear simultaneously. These calculations is done by taking each n-word in each sentence in sequence and find word that occur together more often as word that have a deeper meaning.
3. Calculate Similarity between Terms
This process is a major part from this search engine. The calculation of similarity is to find the relation between terms. The determination of similarity is based on the closeness of the vector of a term with another term in a vector space. The dimensions of this term is determined from the frequency of a term appears together with other

terms. To speed up the process and improve the outcome, this process required term removals that have too little relationships with the other terms.
4. Calculate Document Dimension
The calculation of each dimension of documents is done by multiplying the term frequency (TF), inverse document frequency (IDF), and term vectors.
5. Calculate Similarity between Document and Keyword
In this process, every document and the keyword is considered a vector in vector space and the similarity will be search by comparing the vector space.

6 SYSTEM IMPLEMENTATION AND TESTING

We try to implement this search engine using research document in English and Indonesian language (in pdf format), but this system also can process documents in txt. First enter the document into the system (Figure 4).

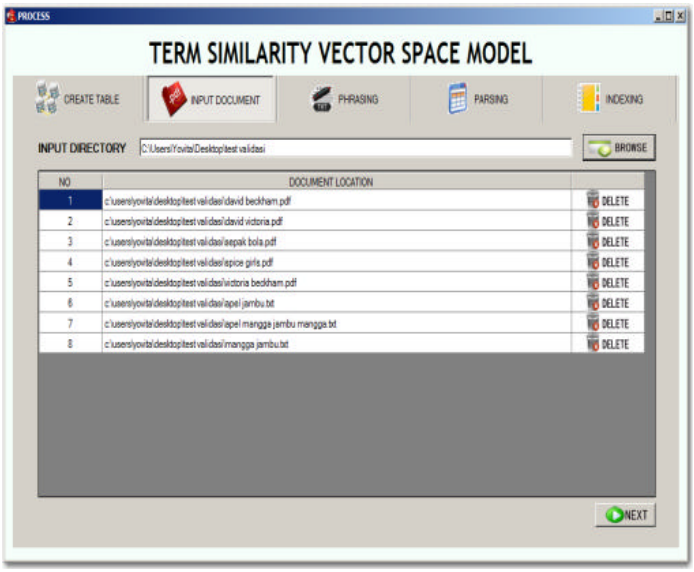


Fig. 4. Document Input Page

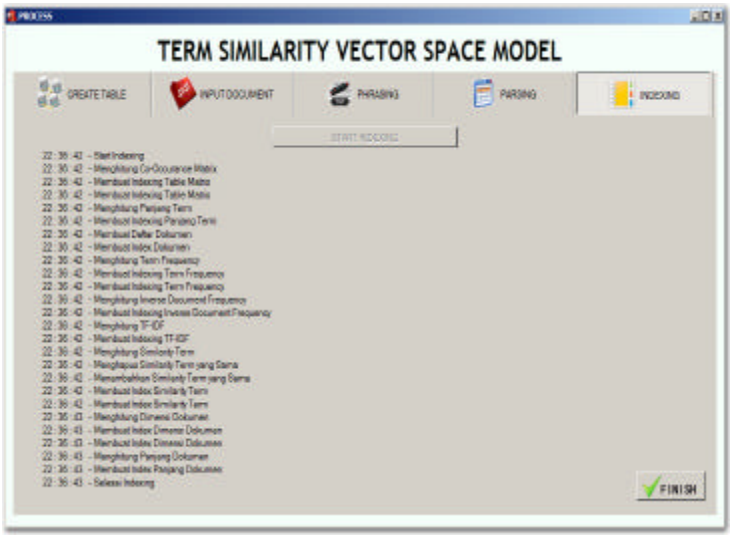


Fig. 5. Indexing Page

Once the documents are inserted, the process of phrase extraction, parsing, and indexing can be done (Figure 5), this will consume a little time. After that searching process is done by entering the keyword that user looking for (Figure 6), the search results will display the documents that have a correlation (on the right side) which also has a link to the document and the terms that have a correlation (on the left side).

Fig. 6. Process Result Page

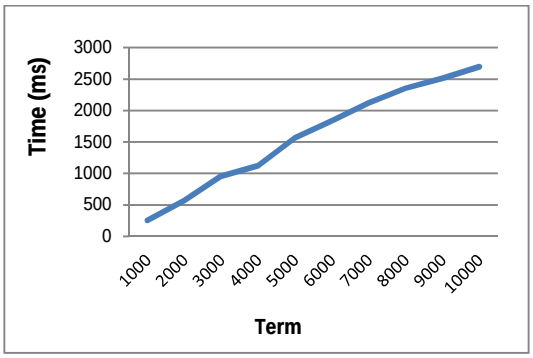


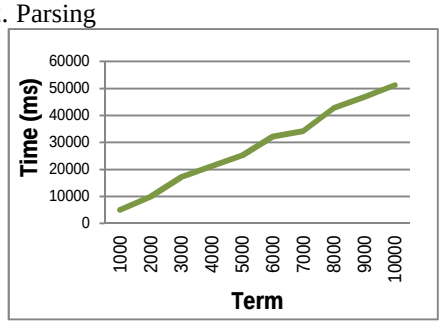
Fig. 7. Amount of Term vs Time on Phrase Extraction

The following figures are the results from our experiment on TVSM.

1. Phrase Extraction

As seen on figure 7, the increasing number of terms versus length of time on phrase extraction O(0.28n+71). While the increasing size of the file versus time O(46.63n+40.19).

2. Parsing



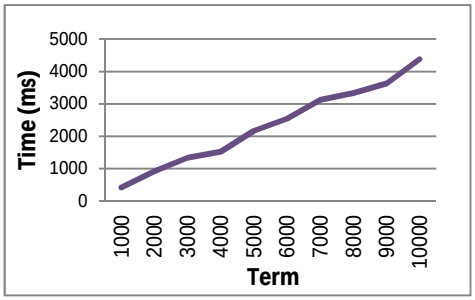


Fig. 8. Amount of Term vs Time on Parsing

Fig. 9. Amount of Term vs Time on Indexing

As seen on figure 8, the increasing number of terms versus parsing time O(5.15n+225). While the increasing size of the file versus time O(864.97n-403).

3. Indexing

As seen on figure 9, the addition amount of term versus indexing time O(0.42n+15.5). While the increasing size of the file versus time O(70.87n-30.535).

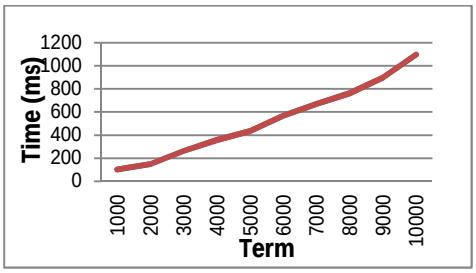


Fig. 10. Amount of Term vs Time on Searching

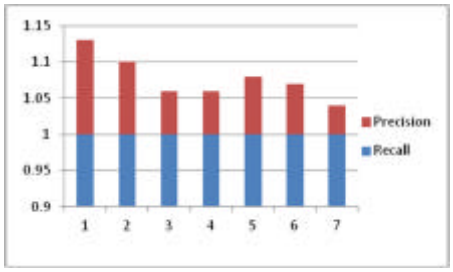


Fig. 11. Precision vs Recall

The increasing number of terms, the more time that required on searching document O(0.11n-63). As shown on figure 10. While the increasing size of the file versus time O(18.04n-74.2). While the results of tests on recall and precision of 7 keywords on the 100 documents (figure 11). It appears that search engines have a maximum recall and a reasonable precision.

7 CONCLUSION

Based on the testing results that performed on the system it can be concluded that Term Similarity Vector Space Model can provide to produce documents that not contain the keyword but contains other keywords that are related. The Indexing process consume a lot of time, because this algorithm will be compares the relationship between terms based on term occurrences in a paragraph. Based on testing of the searching process, the more terms there are, the more time it takes to increase linearly. The searching process for document in Indonesian language is only takes a little bit more time than English language document, especially on the stemming process, but not too significant.

8 REFERENCES

1. Citeseerx. citeseerx.ist.psu.edu/index. Retrieved September 20, 2014 from http://citeseerx.ist.psu.edu/index. (2014).
2. Wibowo, A., Handojo, A, Halim, A. Application of Topic Based Vector Space Model with WordNet. International Conference on Uncertainty Reasoning and Knowledge Engineering (2011).
3. The Apache PDFBox. A Java PDF Library. Retrieved October 2, 2014 from https://pdfbox.apache.org/ (2009).
4. Karaa, WBA. And Gribaa, N. Information Retrieval with Porter Stemmer: a New Version for English. Adv. in Comput. Sci., Eng. & Inf. Technol., AISC 225, pp.243-254, Springer International. (2013).
5. Kikui, Genichiro. Term-list translation using mono-lingual word co-occurrence vector. Japan: NTT Information and Communication System Labs. (1998).
6. Baeza, Y. R. and Frakes, W. B. Information retrieval data structures and algorithms. Upper Saddle River: Prentice Hall PTR. (1992).

7. Grossman, D. and Freider, O. Information retrieval: Algorithms and heuristics (2nd Ed.). Netherland: Springer. (2004).
8. Davis, J. and Goadrich, M. The Relationship Between Precision-Recall and ROC Curves. The 23rd. International Conference on Machine Learning, Pittsburgh, PA. (2006).
9. Ikvm .net home page. Ikvm .net. Retrieved September 20, 2014 from http://www.ikvm.net/. (2014).