LATENT SEMANTIC ANALYSIS AND WEIGHTED TREE SIMILARITY FOR SEMANTIC SEARCH IN DIGITAL LIBRARY

Umi Sa’adah¹, Rianarto Sarno², Umi Laili Yuhana¹
¹Informatics Department, Electronic Engineering Polytechnic Institute of Surabaya
²,³Informatics Department, Faculty of Information Technology
Institut Teknologi Sepuluh Nopember Jl. Raya ITS Sukolilo, Surabaya 60111- Indonesia
email : umi@eepis-its.edu¹, rianarto@if.its.ac.id², yuhana@its-sby.edu¹

ABSTRACT

Digital Libraries grow rapidly in the number of collections and the riches of content. Full text search has linguistic weaknesses, i.e. cannot find documents that have similar meanings/synonyms. This weakness can be overcome by the semantic search by understanding the purpose of search and contextual meaning of the term as shown in the data search.

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual meaning of the word. LSA approach treats the data association between words and documents are examined as a matter of statistics. The goal is to obtain an effective model in representing the relationship between words and documents.

This research applies a search using a combination of the LSA method and weighted tree similarity algorithm. The goal is to combine the search based on metadata and semantic-based search. Users enter a query in the form of a keyword that represents the desired book content. The process is done in two phases, namely the preparation phase and matching phase. In the preparation phase, the system reads the table of contents of entire books in the database to be processed LSA. The process produces the document scores of each title that will become the keyword branch local similarity values of book’s weighted tree. After that, matching process is then performed between the book’s weighted trees with weighted tree of the user query. The system will respond with a list of book titles along with the similarity value based on the value of doc scores.

Keywords: LSA, weighted tree, weighted tree similarity, digital library.

1. INTRODUCTION

To cover a large number of documents that have a very varied content, it takes a search method that can effectively and efficiently provide fast and relevant results [1]. There are several search techniques that have been developed, including full text search, search with metadata, and search with semantic networks.

Full text search matches keywords to each document. The result is a very long list. Full text search has a disadvantage in finding a synonym and homonym [2].

Metadata search is better than full text search. Metadata means data from the data, which is an organized collection of words with “AND” and “OR” logic. Each document is indexed and made its metadata. Search keywords will be matched with the metadata that has been formed with certain restrictions so that the search process much simpler [3]. As well as full text search, metadata search has a weakness in terms if search words that have synonyms and homonym.

The method offered in the search process to overcome these weaknesses is the semantic search. In this research, the model was constructed by calculating the similarity search using the combination of LSA method and Weighted Tree similarity algorithm. In the theory, LSA is able to find keyword semantic relation that is language independent. The research was implemented in the digital library collection of books in Indonesian language. The system will respond by providing relevant search results to documents in Indonesian language.

2. THEORETICAL BACKGROUND, DESIGN AND IMPLEMENTATION

2.1 Theoretical Background

2.1.1 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual meaning of the word. This is done through statistical calculations applied to the corpus of text documents. The basic idea is that the
overall context of a given word does not appear as a set of constraints. This is the crucial thing in common meanings of words and a set of words to each other [4].

Briefly, LSA is a method in which the singular value decomposition is used to form a generalization of the textual semantics. LSA uses the fact that certain words appear in the same context to establish a relationship between the meanings of the word. This allows parts of the text to be compared with each other more intelligent than compare those words directly. The words that never appear together can be compared with the meaning [5].

**How to Calculate LSA**

The first step to do is to represent the text into a term-document matrix (TDM). In this case, each row represents a unique term and each column represents a document. Then the cells of the matrix filled in the frequency of occurrence term to each document. Then, the TDM will be processed using Single Value Decomposition (SVD) technique [6].

SVD is based on a theorem in linear algebra which states that a square matrix can be broken down into a multiplication of three matrices: an orthogonal matrix \( U \), a diagonal matrix \( S \), and a matrix transpose of the matrix \( V \). Theorem is stated as follows

\[
A_{\text{max}} = U_{\text{max}} \cdot S_{\text{max}} \cdot V^T_{\text{max}}
\]  
(1)

Where \( U^T U = I \); \( V^T V = I \); columns of \( U \) are orthonormal eigenvector of \( A A^T \); columns of \( V \) are orthonormal eigenvector of \( A^T A \), and \( S \) is a diagonal matrix that contains the root of the eigenvalue of \( U \) or \( V \) in the order down [7].

**2.1.2 Weighted Tree Similarity**

Weighted Tree Similarity algorithm is one of algorithms for determining the degree of similarity of two objects is presented in the form of tree. The basic algorithm is to compare the attributes of the two objects in stages through its tree representation form, in which each edge is weighted. This algorithm is used for example in Agent Matcher architecture. For example on a virtual market, this architectural used to find the degree of similarity between the buyer agent and seller agent [8].

**2.1.2.1 Determination of Global Similarities**

Determination of similarity is generally done by determining the distance or the similarity level. Two objects are said to have a high degree of similarity if the value of the distance is close to zero or the similarity value is close to 1. Weighted Tree Similarity Algorithm can be used to determine the similarity between trees. The algorithm works by processing weighted tree and will generate output similarity value between 0 and 1. Similarity value is 0 if the two trees have no similarity at all and a value of 1 if the two identical trees [8].

**2.1.2.2 Determination of Local Similarities**

Local similarity calculation is determined from the similarity between the two node labels. Thus, the determination of value is not necessarily by using the exact string matching. But it can also be done in other ways. It is intended that the value of the label similarity between two nodes is not an absolute value of 0 and 1 only, but could be worth a decimal between 0 and 1 [9].

In this research, the calculation of the similarity between the two branches of the label string value is calculated using cosine similarity measure algorithm. Through this algorithm is expected to be obtained similarity value between 0 and 1 between two strings that have the same parts

**2.2 DESIGN AND IMPLEMENTATION**

In general, the design of the system used is shown in the diagram contextual system in Figure 1. Administrator performs the preparation phase until the completion of the SVD decomposition process. Matching phase is initiated by a user who performs the search process by entering a keyword query in the form desired. The output of this system is a list of books in accordance with the user query and the similarity value.
2.2.1 Preparation Phase

Process steps in the preparation phase is shown in Figure 2.

A. Pre Processing

The initial phase begins with the indexing process to create a term document matrix (TDM) from table of contents is read from the database.

The indexing process is performed on all the records in the database of books that aims to generate significant terms of each document. Furthermore, the term document matrix composed of rows represent the terms of all the documents that have been successfully indexed, while the columns represent the document/table of contents that has been read. Elements of the matrix are the frequency of occurrence of a corresponding term in the document, as shown in Figure 3.

B. LSA process by SVD

TDM becomes the input to the LSA process with Single Value Decomposition (SVD). SVD is based on a theorem in linear algebra which states that a square matrix can be broken down into a multiplication of three matrices: an orthogonal matrix U, a diagonal matrix Σ, and an orthogonal matrix transpose of the matrix V as in equation (1). SVD operation will generate three matrices as outputs, which are scales matrix, matrix of term vectors and matrix of doc vectors. In this research, SVD operation is utilizing the existing SVD class in the LingPipe package.
2.2.2 Matching Phase

Process steps in the matching phase are shown in Figure 4.

A. Compute the Query Vector and Doc Scores

This process begins by calculating the value of the query vector to calculate the closeness of a query with a document. This calculation is performed by comparing the keyword query into the value of term vectors. The next process is calculating value of document scores that are the result of the dot product of scales, the query vector, and the doc vectors.

The doc scores values are the basis for determining the context similarity between the keyword with the desired book. The value given to the keyword branch of a book tree, so that it becomes a local similarity value for the branch.

B. Build a Weighted Tree

At this stage, the process is done by generating trees of all books in database using XML code. Furthermore, it also generates XML code for weighted tree based on all the input data and the weights on the user query.

Figure 4. Matching Phase

This phase is initiated by a user who performs a query by entering a keyword from the desired book. Users enter data and gave a weight to each data item.

Figure 5. Books Database Collection
C. Compute the weighted tree similarity
This process is done in two stages:
1. Calculate the local similarity
   - For the branch of title, publisher, and author who has a string data type, then use the Cosine Similarity Measure algorithm to get the value of local similarity.
   - For the branch of year and edition, then use string matching to obtain its similarity
   - For the branch of keyword, do scalar multiplication among the value of scales, query vectors and doc vectors to get the doc scores as the similarity value.
2. Calculate the global similarity
   - Compare each tree of books with tree of user query to obtain the similarity value of each book.
   - Get the ranking of the results of the similarity calculation
   - As output, show the sequence number, the title of the book along with the similarity value
The book that has the greatest similarity value will be ranked the highest, which means the book has the greatest relevance to user queries

3 RESULT
Search process is performed on a collection of books database that shown in Figure 5.
Search model with a combination of weighted tree similarity algorithm with LSA method. This model consists of two phases, namely the preparation phase and phase matching

3.1 Preparation Phase
In this phase, get the scale and term vector values from all books in the database collection. The results are shown in Figure 6 and Figure 7

3.2 Matching Phase
In this phase the user enter the query from the user interface as shown in Figure 8. For example, the user gives the user query with title "Database Relasional" and keyword "relasional". In this case, the user give for all branches the value of weight by 25, except for the keyword branch that the weight value is 100.
Furthermore, calculate the local similarity for the title, author, publisher, year and edition. LSA was applied to the keyword branch. Its local similarity is computed with LSA method to find the value of doc scores.

After calculating each local similarity, the next process is the computation of global similarity with weighted tree similarity algorithm. The result is a list of books with its similarity value as shown in Figure 9.

In the output, the system recommends some titles that match the user query. Books with titles containing the word "Relational Database" ranked first. The next ranking are books that are similar to a given query context, i.e. books that relate to relational database. In this experiment, the top 10 recommended books all related to relational databases as shown in Figure 9.
4 CONCLUSION

Search system using combination of weighted tree similarity algorithm and LSA method on a collection of books in Indonesian language capable of providing relevant results. This is evident from the similarity context between the keywords entered and the recommended book’s titles.

REFERENCE


