Int. Jour. of App. Sc. and Eng. (IJASE) **2**(2) : Dec 2014, 111-116


# A dynamic replica recognization using lsi algorithm

R.Thiyagarajan[1] and T.K.P.Rajagopal[2]

[1]II M.E (CSE), Kathir College of Engineering, Coimbatore, Tamil nadu, India.
[2]Head of the Department(CSE), Kathir College of Engineering, Coimbatore, Tamil nadu, India.

Corresponding author:

**Abstract**

Many data mining techniques have been proposed for mining useful patterns in text documents. However, how to effectively use and update discovered patterns is still an open research issue, especially in the domain of text mining. Since most existing text mining methods adopted term-based approaches, they all suffer from the problems of polysemy and synonymy. Over the years, people have often held the hypothesis that pattern (or phrase)-based approaches should perform better than the term-based ones, but many experiments do not support this hypothesis. This project presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information. For efficient pattern mining, Latent Semantic Indexing (LSI) is used, a new approach to automatic indexing and retrieval. It is designed to overcome a fundamental problem that plagues existing retrieval techniques that try to match words of queries with words of documents. The particular "latent semantic indexing" (LSI) analysis that we have tried uses singular-value decomposition. We take a large matrix of term-document association data and construct a "semantic" space wherein terms and documents that are closely associated are placed near one another. Singular-value decomposition allows the arrangement of the space to reflect the major associative patterns in the data, and ignore the smaller, less important influences. As a result, terms that did not actually appear in a document may still end up close to the document, if that is consistent with the major patterns of association in the data. Position in the space then serves as the new kind of semantic indexing, and retrieval proceeds by using the terms in a query to identify a point in the space, and documents in its neighborhood are returned to the user.

**Keywords:** Replication, Similarity, Latent Semantic Indexing.

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize

the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

A process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can learn more about their customers and develop more effective marketing strategies as well as increase sales and decrease costs. Data mining depends on effective data collection and warehousing as well as computer processing.

### *Duplication*

In real time applications, identification of records that represent the same real world entity is a major issue to be solved. Such records are termed to be duplicate records. The duplicate detection is an important step for data integration.

### *Pre-Duplicate Record Detection Phase*

Detection and removal of duplicate records that relate to the same entity within one data set is an important task in case of the date preprocessing. Data linkage and duplication can be used to improve data quality and integrity, to allow re-uses of existing data sources for future research work.

### *Data Processing*

In real-world, data tend to be incomplete, noisy and inconsistent. Such situation requires data preprocessing. Various forms of data preprocessing includes data cleaning, data integration, data transformation and data reduction. In other words, the data preparation stage includes data cleaning, data transformation and data standardization. Typically, the process of duplicate detection is preceded by a data preparation stage, during which data entries are stored in a uniform manner in the database. Data cleaning process attempts to fill the missing values, smooth out noise while identifying outliers and correct inconsistencies in the data. Data transformation process converts the data into appropriate forms for mining. Data reduction techniques can be used to obtain a reduced representation of the data while minimizing the loss of information content.

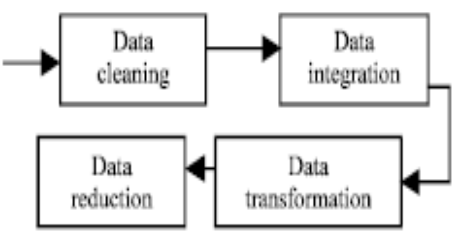


**Figure 1. Steps in data preprocessing**

*Techniques to Match Individual Fields*

The typographical variations of string data is one of the most common (sources) reasons of mismatches in database entries. Hence, duplicate detection typically relies on string comparison techniques to deal with typographical variations. Based on various types of errors, multiple methods have been developed for this task namely:

- Character-based similarity metrics
- Token-based similarity metrics
- Phonetic similarity metrics
- Numeric similarity metrics

These methods can be used to match individual fields of a record. In most real-life situations, records consist of multiple fields. Thus, (record) duplicate detection problem becomes more complicated. There are two (methods) categories used for matching records with multiple fields namely:

Probabilistic approaches and supervised machine learning techniques.

- Usage of declarative languages for matching and devise distance metrics for duplicate detection task.

**Document Comparison**

This module gets the document from the user for e.g. :( Document 1 and Document 2), the input document is been compared with the existing document. The document comparison acts as two functions such as content comparison and polysemy, a dataset is collected for the operation of polysemy.


**Figure 2: Comparing two documents**

**Term Document Matrix Generation**

In term document, matrix generation is obtained by the data from the input document s and the

collected data set, which are gathered from document comparison module. If the comparison terms are same we assign matrix value as 1, if differs the value will be 0.

### Decompose Matrix and Ranking

Decompose matrix A matrix and find the U, S and V matrices, where implement a Rank 2 Approximation by keeping the first two columns of U and V and the first two columns and rows

### Similarity Calculation

The similarity calculation is done from the ranking results generated from the decomposes matrix and according to the values generated and the results are obtained successfully, In such a way that according to the result it calculate whether it is duplicate or not.

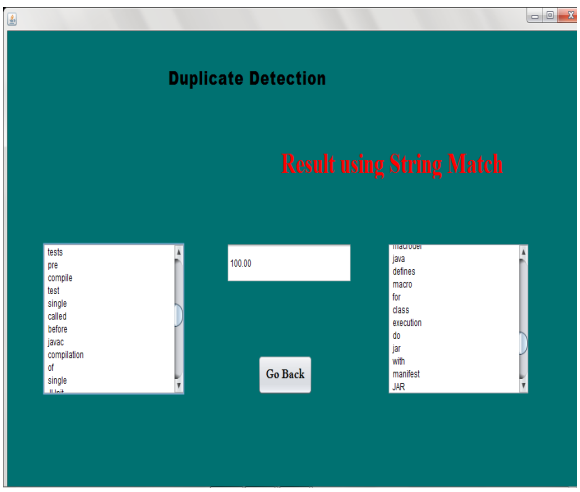


**Figure 3. String match between document 1 and document 2**

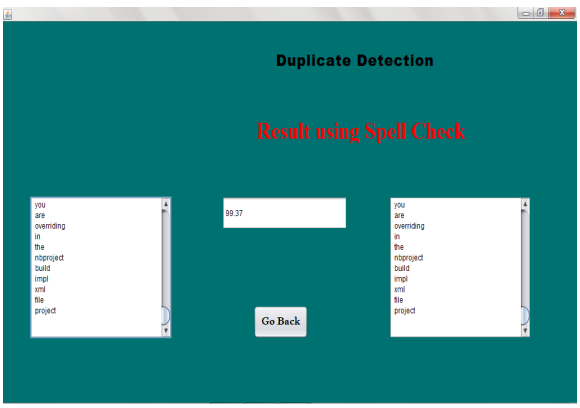

**Figure 4. Spell checking between document 1 and document 2**

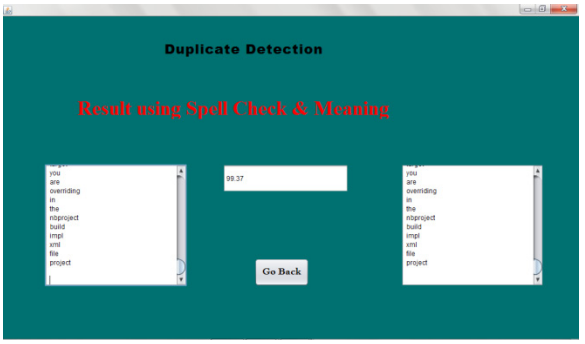


**Figure 5. Checking the meaning between document 1 and document 2**

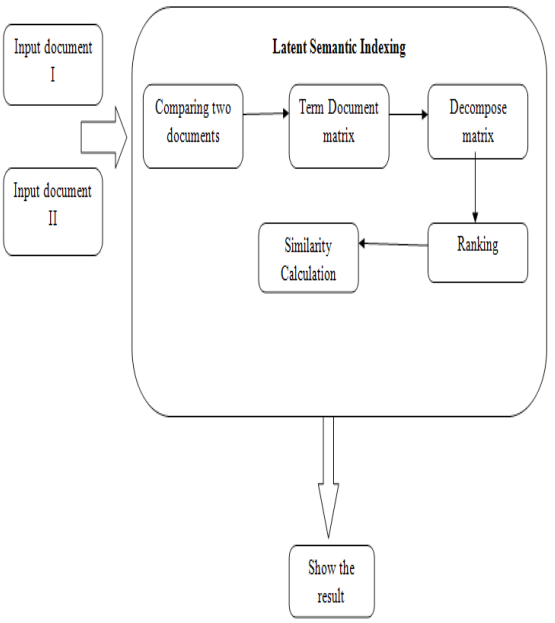


**Figure 6. Architecture Diagram**

## Conclusion

Our project presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information. For efficient pattern mining, Latent Semantic Indexing (LSI) is used, a new approach to automatic indexing and retrieval. The particular "latent semantic indexing" (LSI) analysis that we have tried uses singular-value decomposition. We take a large matrix of term-document association data and

construct a "semantic" space wherein terms and documents that are closely associated are placed near one another. Singular-value decomposition allows the arrangement of the space to reflect the major associative patterns in the data, and ignore the smaller, less important influences. The technique decomposes the given document and find similarity between two documents based on ranking.

## References

[1] A New Model for Secure Dissemination of XML Content by Ashish Kundu, Student Member, IEEE, and Elisa Bertino, Fellow, IEEE.

[2] R.Ananthakrishna, S. Chaudhuri, and V. Ganti, "Eliminating Fuzzy Duplicates in Data Warehouses," Proc. Conf. Very Large Databases (VLDB), pp. 586-597, 2002.

[3] An Efficient Duplicate Detection System for XML Documents by Thandar Lwin and Thi Thi Soe Nyunt.

[4] An Efficient Duplicate Image Detection Method Based On Affine-Sift Feature by Yudong Cao 1,2, Honggang Zhang t, Yanyan Gao t, Jun Guo

[5] Detecting Duplicates in Complex XML Data by Melanie Weis, Felix Naumann

[6] Discovery of Complex Glitch Patterns: A Novel Approach to Quantitative Data Cleaning by Laure Berti-Equille, Tamraparni Dasu, Divesh Srivastava

[7] Eliminating Duplicates in Information Integration: An Adaptive, Extensible Framework by Hamid Haidarian Shahri, Saied Haidarian Shahri

[8] Fuzzy Duplicate Detection on XML Data by Melanie Weis

[9] Intelligent Dynamic XML Documents Clustering by Laura Irina Rusu, Wenny Rahayu and David Taniar

[10] Intelligent Dynamic XML Documents Clustering by Laura Irina Rusu, Wenny Rahayu and David Taniar

[11] D.V. Kalashnikov and S. Mehrotra, "Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph." ACM Trans. Database Systems, vol. 31, no. 2, pp. 716-767, 2006.

[12] A.M. Kade and C.A. Heuser, "Matching XML Documents in Highly Dynamic Applications," Proc. ACM Symp. Document Eng. (DocEng), pp. 191-198, 2008.

[13] L. Leita ˜o, P. Calado, and M. Weis, "Structure-Based Inference of XML Similarity for Fuzzy Duplicate Detection," Proc. 16th ACM Int'l Conf. Information and Knowledge Management, pp. 293-302, 2007.

[14] D. Milano,M. Scannapieco, and T. Catarci, "Structure Aware XML Object Identification," Proc. VLDB Workshop Clean Databases (CleanDB), 2006.

[15] F. Naumann and M. Herschel, An Introduction to Duplicate Detection. Morgan and Claypool, 2010.