International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Contextual Plagiarism Detection Using Latent Semantic Analysis

Rajkumar Kundu, Karthik. K
Department of Computer Science, Christ University, Bangalore, Karnataka, India-560029

Abstract**—** *Plagiarism means to commit literary theft, where in one steals ideas or words from another and claim it as their own without crediting the source. There are mainly two types of plagiarism possible; text based similarity and context based similarity. Text based similarity works on a word by word match algorithm. Context based similarity tries to find a match between two series of text with respect to implemented meaning or ideas implied. Latent Semantic Analysis(LSA) is a method which tries to find deeper correlation between words used in a document, here semantic implies that the documents try to find a root topic under which both the documents hold true. In this paper LSA is used for context based similarity between documents.*

Keywords**—** *Contextual plagiarism; latent semantic analysis: vector space model.*

I. INTRODUCTION

Plagiarism is now becoming a well-known problem in academic integrity, especially it is affecting the student's area in terms of gaining the academic knowledge in a proper way. It is actually an illegal constraint that can become a major issue in anyone's personal life. It occurs when someone uses other's work and getting the credit as their own contribution about the work without having the knowledge of the actual author. For a given document, the main task of a plagiarism detection system is to find that particular document (a text doc, an assignment from students or any document in a text format) is copied from any other document or not. There are several techniques for detecting this. For a smart person a normal plagiarism detection system will not be able to detect plagiarism correctly.

There are two major approaches available to develop such kind of system, intrinsic and extrinsic. In extrinsic approach of detection uses several techniques to find the similarities between the source documents and an input document. Here in this approach every document is represented as an n-dimensional vector where n is the number of terms presents in that particular document or some derived features from the document. A number of mathematical measures are available to do the task of comparison between the vectors, by calculating the Euclidian Distance, Cosine Similarity, String Matching and SVD (singular value decomposition).On the other hand in intrinsic approach of detection, each and every source document is analyzed using different techniques without considering the input document. Assuming that a good-enough writing style analysis is available, this approach can effectively detect heavy-revision plagiarism cases or even plagiarism cases from a different language (multi-lingual plagiarism).

In this paper the extrinsic approach has been used where each source document running under the pre-processing task, where all stop words presents in the document has to be removed first and then it performs stemming for decreasing the calculative overload. This proposed conceptual plagiarism detection approach is able to filter the observed unique word from the collection of source documents and map them in an n-dimensional semantic space. Next the local input document is used to map its observed words in that same semantic space. The Cosine similarity between two vectors in that semantic space (local observed vector and each documents considered as an individual vector in that space) shows the similarity of observed documents with each individual documents.

II. LATENT SEMANTIC ANALYSIS

Latent Semantic Analysis (LSA) is a method [5] to capture the semantic structure of documents based on patterns of word co-occurrences within texts. The method gets the name latent because it mines deeper correlations among words in the texts that are otherwise unseen. The word semantic implies that the words in a document help to identify the topics or concepts of the document. LSA is a mathematical model that is independent of any sort of external sources of semantics like vocabularies, dictionaries, grammar, syntactic parsers, or morphologies [6]. LSA uses Singular Value Decomposition (SVD) followed by dimensionality reduction to capture all correlations latent within documents by modeling interrelationships among words so that it can semantically cluster words and documents that occur in similar contexts.

Singular Value Decomposition

SVD is the core process of LSA. It is a technique in linear algebra for matrix decompositions that breaks down a matrix A into three matrices U, S, and V. According to the theorem mentioned by Baker [7], a rectangular matrix A_{mn} of order m× n can be broken down into the product of three component matrices—an orthogonal matrix U_{mm}, a diagonal matrix S_{mn}, and the transpose of an orthogonal matrix V_{nn}. The theorem is usually presented as follows:

$$A_{mn} = U_{mm}S_{mn}V^{T}_{nn},$$

where $U^TU = I$, $V^TV = I$, with I being an identity matrix, the columns of U and V are orthonormal eigenvectors of AA^T and A^TA, respectively, and S is a diagonal matrix containing the square roots of eigenvalues from U or V, known as singular values, sorted in descending order.

Dimensionality Reduction

In SVD, a matrix is broken down into three component matrices using Eigen decomposition in a way that the product

Rajkumar Kundu and Karthik. K, "Contextual plagiarism detection using latent semantic analysis," *International Research Journal of Advanced Engineering and Science*, Volume 2, Issue 1, pp. 214-217, 2017.

of these three component matrices reconstructs the original one. Within the context of LSA, this reconstruction is only an approximation of the original matrix based on a reduced number of dimensions of the component matrices.

Mathematically, the original representation of data in matrix A_{mn} is reconstructed as an approximately equal matrix Ak_{mn} from the product of three matrices U_{mk}, S_{kk}, and V^{T}_{kn} based on just k dimensions of the component matrices U_{mm}, S_{mn}, and V_{nn} of the original matrix A_{mn}. The diagonal elements of matrix S are non-negative descending values. When S is reduced to a k×k order diagonal matrix S_{kk}, the first k columns of U_{mm} and V_{nn} form matrices U_{mk} and V_{nk}, respectively. The reduced model is

$$Ak_{mn} = U_{mk}S_{kk}V^{T}_{kn}$$

This approximate representation of the original documents after dimensionality reduction reflects all underlying correlations. Words that occurred in some context prior to dimensionality reduction now become more or less frequent, and some words that did not appear at all originally may now appear significantly or at least fractionally. This lower-dimensional matrix representation of the linguistic texts is termed *semantic structure*, *LSA space*, or *semantic space* the literature [5], [6]. Within this structure, terms that never occurred in a document become related to each other, thereby bringing out the latent semantic structure of the vocabulary used in the document collection.

Document Classification Using LSA

LSA is a proximity model that spatially groups similar points (words and documents) together. Through a combination of SVD and dimensionality reduction, points that occur in similar contexts become more similar, moving closer to the concepts in the semantic space. As the dimensional space is reduced, related points draw closer to one another. The relative distances between these points in the reduced vector space show the semantic similarity between documents, which is used as a basis for document classification. The initial VSM term space representing the training set of documents is used to derive the LSA semantic space. The test document that is to be classified is mapped as a pseudo-document into the semantic space by the process of "folding-in" [8]. Then the pseudo-document's closeness with all other documents is measured using any of the standard measures of similarity, such as cosine measure and Euclidean distance. The category of the document that is located in its nearest proximity in space is the category of the test document.

III. Literature Survey

Nowadays, learning and reasoning from multiple documents requires students to employ the skills of sourcing (i.e., attending to and citing sources) and information integration (i.e., making connections among content from different sources). Source's Apprentice Intelligent Feedback mechanism (SAIF) is a tool for providing students with automatic and immediate feedback on their use of these skills during the writing process. SAIF uses Latent Semantic Analysis (LSA) [1], a string-matching technique and a pattern-matching algorithm to identify problems in students' essays. These problems include plagiarism, uncited quotation, lack of citations, and limited content integration. SAIF provides feedback and constructs examples to demonstrate explicit citations to help students improve their essays. In addition to describing SAIF, two experiments are done [1]. In the first experiment, SAIF was found to detect source identification and integration problems in student essays at a comparable level to human ratters. The second experiment tested the effectiveness of SAIF in helping students write better essays. Students given SAIF feedback included more explicit citations in their essays than students given sourcing-reminder instructions or a simple prompt to revise.

Also plagiarism is an increasing problem in the digital world [3]. The sheer amount of digital data calls for automation of plagiarism discovery. This paper evaluates an Information Retrieval approach of dealing with plagiarism through Vector Spaces. This will allow us to detect similarities that are not result of naive copy and paste. We also consider the extension of Vector Spaces where input documents are analyzed for term co-occurrence, allowing us to introduce some semantics into our approach beyond mere word matching. The approach is evaluated on a real-world collection of mathematical documents as part of the DML-CZ project [3]. The goal in information retrieval [9] is to enable users to automatically and accurately and data relevant to their queries. One possible approach to this problem is to use the vector space model, which models documents and queries as vectors in the term space. The components of the vectors are determined by the term weighting scheme, a function of the frequencies of the terms in the document or query as well as throughout the collection. Term weighting schemes are broadly classified into two categories: unsupervised and supervised [10].

IV. Dataset and Pre Processing

A data set is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set.

A. Dataset

The PAN plagiarism corpus 2011 (PAN-PC-11) is a corpus for the evaluation of automatic plagiarism detection algorithms.

The PAN-PC-11 contains documents in which plagiarism has been inserted automatically as well as documents in which plagiarism has been inserted manually. The former have been constructed using a so-called random plagiarist, a computer program which constructs plagiarism according to a number of parameters. We use only 300 textual pieces in our application. For the experiments, we randomly select 1 record from allotted input set. The training set consists of 60 records. After constructing latent semantic space with 240 records as a

Rajkumar Kundu and Karthik. K, "Contextual plagiarism detection using latent semantic analysis," *International Research Journal of Advanced Engineering and Science*, Volume 2, Issue 1, pp. 214-217, 2017.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

training dataset, we randomly select 1 record that consists of test dataset to check the similarity.

TABLE I. Dataset.

Document Attributes	Values
Number of documents in our dataset	300
Number of documents in training set	240
Number of documents in test set	60

B. Pre Processing

Data pre-processing is an important step in the data mining process. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data pre-processing includes cleaning, normalization, transformation, feature extraction and selection, etc. The product of data pre-processing is the final training set. In our application the data pre-processing is done in two steps. They are stop words removal and stemming.

Stop words are words which are filtered out before or after processing of natural language data. Though stop words usually refer to the most common words in a language, there is no single universal list of stop words used by all natural language processing tools, and indeed not all tools even use such a list. Some tools specifically avoid removing these stop words to support phrase search.

Any group of words can be chosen as the stop words for a given purpose. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as "The Who", "The The", or "Take That". Other search engines remove some of the most common words—including lexical words, such as "want"—from a query in order to improve performance. Next Stemming operation has been performed on those observed words.

Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. A stemmer for English, for example, should identify the string "cats" (and possibly "catlike", "catty" etc.) as based on the root "cat", and "stems", "stemmer", "stemming", "stemmed" as based on "stem". A stemming algorithm reduces the words "fishing", "fished", and "fisher" to the root word, "fish". On the other hand, "argue", "argued", "argues", "arguing", and "argus" reduce to the stem "argu" (illustrating the case where the stem is not itself a word or root) but "argument" and "arguments" reduce to the stem "argument".

V. METHODOLOGY AND IMPLEMENTATION

LSA is categorizes semantically related texts as similar even when they do not share a single term. This is because in the reduced semantic space, the closeness of documents is determined by the overall patterns of term usage. So documents are classified as similar regardless of the precise terms that are used to describe them. As a result, terms that did not actually appear in a document may still end up close to it if that is consistent with the major patterns of association in the data.

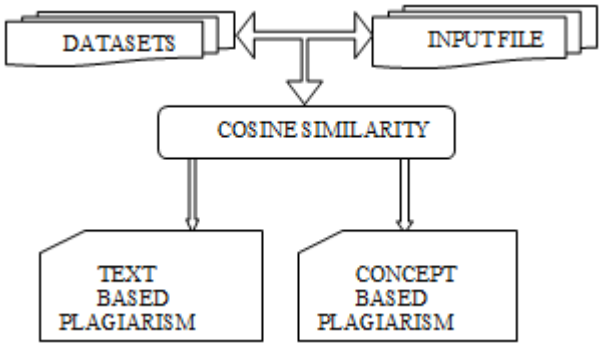

Fig. 1. System architecture.

Figure 1 shows about the system architecture demonstrates that the datasets are divided into train and test data. After the model is constructed for the train data, the model is used to find the similarity of the test data. The similarity is found using the cosine similarity. Although we obtain both text based and concept based similarity.

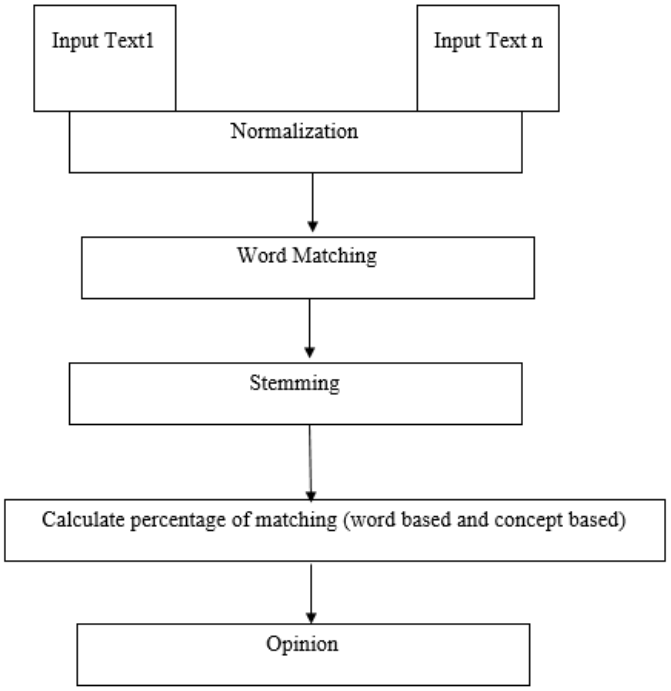

Fig. 2. System model.

Figure 2 represents the system model demonstrates the actual flow of our application to implement the knowledge. The model is used to find the similarity of the test data among 240 trained dataset.

Rajkumar Kundu and Karthik. K, "Contextual plagiarism detection using latent semantic analysis," *International Research Journal of Advanced Engineering and Science*, Volume 2, Issue 1, pp. 214-217, 2017.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

VI. Evaluation an Result

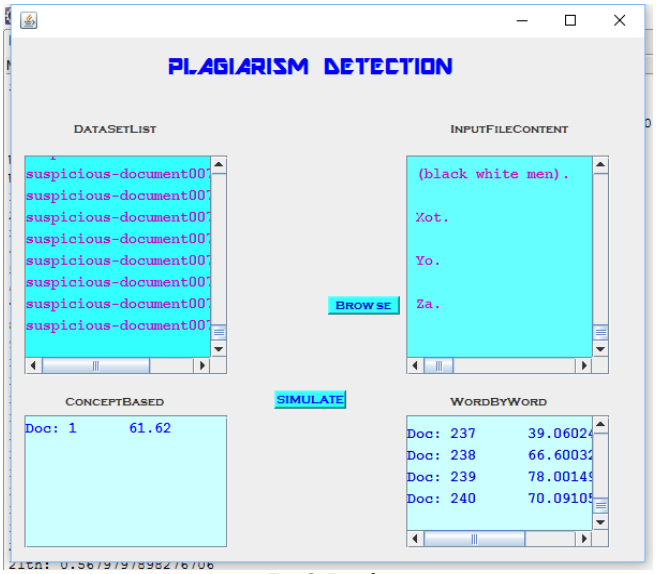

Fig. 3. Result.

Here is the proposed interface system shown in figure 3 created by java applet in Eclipse platform. The test generated with 300 text file where 240(80%) documents considered as training set and remain 60(20%) documents as testing set. The input file has been taken by clicking the "Browse" button and displayed it to "InputFileContent". The actual result will generate by "Simulate" button, where it will consider "InputFileContent" to generate word-count matrix. And pseudo measure of "InputFileContent" compares with the semantic space matrix to compute the similarity. "conceptBased" text area displays concept based similarity between the documents and "WordByWord" text area displays word based similarity between the documents. For each document in the 60 testing set, at least one match of conceptually similar document has been found.

VII. Conclusions and Future Scope

Detection of plagiarism and hence its prevention is a very laborious work that requires deep research of the subject. This project aimed to develop a plagiarism detection system that detects the extent of plagiarism in a particular document uploaded by the user. Subsequent numbers of literatures (Training data set) were reviewed before starting the project. Design considerations were then carefully undertaken and implemented. The result obtained by implementing different algorithms and methods are within the desired framework. Different algorithms and methods are used and the result is shown as desired. The developed system is also compared with the existing plagiarism detection system. Large amount of knowledge has been gained throughout the project work. The importance of the background research, requirement analysis and specifications, well designing concept, and superior methodology were learnt. Also implementation techniques, testing, error handling, optimization issues and the predictability of problems such as when to perform a certain task, have been exercised. Thus we hope that the system developed will certainly contribute for the plagiarism detection and prevention and will be supported by many Free/Open source enthusiasts for its enhancement in the future.

References

[1] K. Krishnamurthi, V. R. Panuganti, and V. V. Bulusu-"Influence of supplementary information on the semantic structure of documents," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, issue 7, pp. 219-225, 2015.

[2] M. A. Britt, P. Wiemer-Hastings, A. A. Larson, and C. A. Perfetti, "Using intelligent feedback to improve sourcing and integration in students' essays," *International Journal of Artificial Intelligence in Education*, vol. 14, issue 3-4, pp. 359–374, 2004.

[3] R. Rehurek, "Plagiarism detection through vector space models applied to a digital library," In Raslan 2008. 1,. Brno: Masaryk University, 2008. pp. 75-83, 9 pp. ISBN 978-80-210-4741-9.

[4] Dr. E. Garcia, Latent Semantic Indexing (LSI) A Fast TrackTutorial online. Available: http://www.miislita.com/information-retrieval tutorial/svd-lsi-tutorial-1-understanding.html.

[5] S. Deerwester, S. Dumais, G. Furnas, and T. K. Landauer, "Indexing by latent semantic analysis," *Journal of the Association for Information Science and Technology*, vol. 4, issue 6, pp. 391–407, 1990.

[6] T. K. Landauer, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, issue 2–3, pp. 259–284, 1998..

[7] K. Baker 2005. Singular Value Decomposition Tutorial. Retrieved September 22, 2016, from http://www.ling.ohio-state.edu/~kbaker/pubs/Singular_Value_Decomposition_Tutorial.pdf..

[8] M. Berry and S. Dumais, "Using linear algebra for intelligent information retrieval," *SIAM Review*, vol. 37, issue 4, pp. 573–595. 1995.

[9] E. Chisholm and T. G. Kolda, "New term weighting formulas for the vector space method in information retrieval," Report ORNL/TM-13756, *Computer Science and Mathematics Division*, Oak Ridge National Laboratory, 1999.

[10] F. Debole and F. Sebastianit, "Supervised term weighting for automated text categorization," in *Proceedings of the ACM Symposium on Applied Computing*, pp. 784–788. 2003.