IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.7, Special Issue.1, Jan 2019 E-ISSN: 2347-2693

Use of Semantic Search to Enhance the Performance of Plagiarism Detection Tools

A. Das[1*], S.Shaw[2]

[1]Dept. Of CSE&IT, JUIT, Waknaghat, Solan, Himachal Pradesh, India
[2]Dept. Of CSE, Jadavpur University, Kolkata, India
[1]Dept. Of CSE, Jadavpur University, Kolkata, India

[*]*Corresponding Author: arijit.mcse.ju@gmail.com*

Available online at: www.ijcseonline.org

***Abstract*—** Plagiarism is a breach of copyright in the academic world. It has become a serious issue after explosion of digital information as copying has become easier due to huge amount of source but detection has become more difficult. Plagiarism can be of two types: source code plagiarism refers to copying the code from proprietary software and text plagiarism which deals with copying from others text and pretending it as own. There are several tools to detect both type of plagiarism. In this paper we have concentrated mainly on text plagiarism discussed about algorithms used in software available like Turnitin, iThenticate or SafeAssign to detect plagiarism and how NLP techniques and parallel processing can improve them. Mostly all software determine a similarity score for each pair of document and use SCAM (Standard Copy Analysis Mechanism) algorithm to calculate relative measure of overlapping during comparison of common set of words. We have tried to establish how semantic similarity can improve TRUE POSITIVE and TRUE NEGATIVE detection and reduce FALSE POSITIVE and FALSE NEGATIVE detection in our work.

***Keywords*—** Plagiarism, Semantic Similarity, Semantic Search, Turnitin, WordNet, Ontologies, Natural Language Processing

I. INTRODUCTION

The word Plagiarism is a consequent of the Latin word "plagiarius" means "an abductor" or "hijacker".Plagiarism may be considered as infringement of someone else's cerebral property rights. The process used to identify and locate plagiarism in a text or a document is known as plagiarism detection. The type of method used for detecting the plagiarism totally depends on the plagiarism type. Plagiarism activities are increasing day by day specially in research activities [1]. To curb this habit various universities and enterprises have developed software to detect plagiarism. But still there are a lot of space to improve this detection result. One of such prominent problem is detection of paraphrasing or obfuscation plagiarism. In this type of plagiarism original text is modified by reduction, combination, paraphrasing, shortening, reforming, concept specification and concept generalization. Most common tools such as Turnitin or SafeAssign are able to detect only copy paste plagiarism. But when any intelligent writer uses synonyms or translations or changes the words keeping the idea same these tools fail to detect the same. Actually all these approaches of plagiarism are in the domain of natural languages. In this paper we have reviewed various NLP techniques which can be used to handle this problem.

Plagiarism can be categorized into many different types but we will restrict our discussion to academic plagiarism.

Academic plagiarism is the plagiarism happened in the academic domain i.e. in the universities, colleges and workshops, seminars, conferences being organized in the academic institutes or journals, critical reviews, programming code, paper, books published by the academic community. In other words this plagiarism does not affect any business gain but on the other hand as there is loose or absence of intellectual property right practice this plagiarism happens frequently. It hampers the credit of real scholars in the academics. The most widely used software products in the world are: iThenticate, SafeAssign and CrossCheck. Turnitin stores all the papers submitted for checking in database for future reference. SafeAssign gives the user option whether the user is willing to store the paper being submitted in the database of SafeAssign. CrossCheck uses some community approach. If any scholarly publisher is affiliated to CrossRef community they need to hand over their database of scholarly articles which will be used by CrossCheck for plagiarism detection. In return those publishers will be able to use CrossCheck free of charge. There is another Viper tool which works in windows

environment only. It is a standalone application which compares in the self declared library.

Plagiarism of ideas, complex paraphrasing, and plagiarism between multiple languages are detected in any of the above soft wares. In this paper we have reviewed various approaches tried by researchers, scientists or engineers to detect them. These approaches are based on mainly on various NLP techniques using WordNet and Ontologies. Some researchers have also proposed to include machine translation module in the soft wares to detect plagiarism between different languages. But all these qualitative improvement is coming in return of adding extra cost to the software. Our target is to derive something without adding any extra cost.

Next we have discussed our proposed techniques based on semantic similarity measurement. The rest of the paper is organized as follows: Section II discusses related work. Various possible detection methods along with corresponding tools are elaborated in section III. A new method related with semantic search to detect plagiarism in an effective manner has been introduced and discussed in section IV. Finally conclusion is done in section V.

II. RELATED WORK

Researchers have tried to approach the problem of contextual plagiarism detection in various ways. Using Word Net is one of the most common approach. Some of them have used Ontology and enhanced it with Fuzzy Similarity measure. Morphological analysis or syntactical analysis over the result data set of WordNet,use of machine learning algorithm upon detected plagiarism are also some of the approaches. Detection of plagiarism from the citations, deep learning from the bibliography, use of graphs to find the relation between two paragraphs and even use of multidimensional tensors are some of the methods scientists have explored.

Tsatsaronis et al. [1] used Wikipedia as their knowledge base and used WordNet to derive synonymy, hyponymy and hypernymy in the target set of data. Then semantic analysis of text was done to find relation between source and target.
Fernando and Stevenson [2] proposed supervised method to detect paraphrase using WordNet as knowledge base.

Shenoy et al [3] proposed an algorithm to learn ontology from different documents available in the internet using OWL (Web Ontology Language).Learning techniques was farther enriched using WordNet. Then this learning set is applied to detect relations between target and source document.

AI-Shamery and Gheni [4] used WordNet to find synonyms and if the number of synonyms crosses a threshold value it was the proof of semantic plagiarism.

Alzahrani and Salim [5] used shingling and Jaccard coefficient to pull potential source documents. Then fine grain (upto word level) source and target was compared using Fuzzy techniques and WordNet. Three fuzzy degree of similarity was proposed- 1 for exact word match,0.5 for WordNet synonyms and 0 for different words.

Marsi and Krahmer [6] proposed building of syntactical trees of sentences from both target (T) and source (S) documents. Then each node of T was matched with corresponding node of S. They used ML algorithms also to improve the tree building capacity.

Czerski et al. [7] tried to attack the problem from different dimension. Synonyms from the wordnet and Thesaurus and IS A relationship from the ontology was used to replace the words in the target document and thus the number of comparisons were reduced. At last it was only lemma matching.

Eissen and Stein [8, 9] used stylometry analysis to detect similarity. Five classes of stylometric features were used namely 1. structural features, which reflect text organization, 2. closed-class word sets to count special words, 3. part-of-speech features to quantify the use of word classes, 4. syntactic features, which measure writing style at the sentence-level and 5. text statistics, which operate at the character level . Using these 5 classes they derived average word frequency class concept which proves to be one of the most successful technique to detect semantic plagiarism.

Gipp et al. (2013) approached to catch plagiarism based on citation. As per their observations people don't change the cited text so that can be source for semantic plagiarism.They got a considerable success using this approach.

Osman et al. [10] built graph by grouping each sentence term into one node considering both source and target document and also building two graphs. The resultant nodes are connected to each other based on their order in the original document and also a top node is formed according to the concept of the sentence and grouping similar terms to them. All nodes are connected to the top level node also.

Chong et al. [11] used several NLP techniques like word sense disambiguation, POS tagging, Root Verb extraction etc. and also got success to detect the plagiarized text. Though synonymy detection, sentence structure generalization etc remained challenge.

Gharavi et al. [12] worked in the Persian language data set. He represented words as multi dimensional vectors and combined the word vectors to represent a sentence using aggregate function. By comparing vectors formed from source and target document the highest similarity vector is

suspected as suspicious candidate for plagiarism. The evaluation is repeated twice.

Agirre, Eneko et al. [13] used a text to text similarity metric which was formed using two corpus based and six knowledge based methods. Performance wise this outperformed the vector based approach.

Kong et al. [14] used Logical Regression model. The proposed model analyzed suspicious documents and source documents and extracted lexicon, syntax, semantics and structure features which are used as training set of the model.

III. PLAGIARISM DETECTION METHODS AND TOOLS

Plagiarism can be textual as well as source code plagiarism. In case of code plagiarism codes are copied and this case is considered in case of proprietary codes. Code plagiarism happens both in classroom and enterprises. According to the survey 20% of total plagiarism in Stanford University is in Computer Science class. Enterprises are concerned about the code plagiarism to protect their technical expertise and business secret. The fight between Google and Oracle ran for seven years in the court for the ownership of java code used in android app. These cases are the example of code plagiarism. Various tools which are used to detect source code plagiarism are shown below in figure 1.

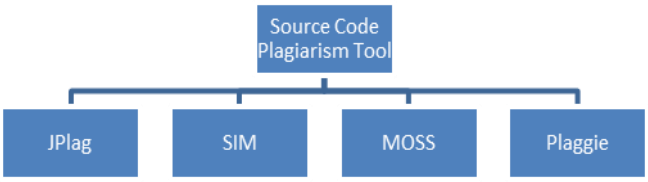

Fig. 1 Source Code Detection Tools

Textual Plagiarism is copying from others without admitting. Text plagiarism happens in natural languages. Text plagiarism detection tools are shown below in figure 2.

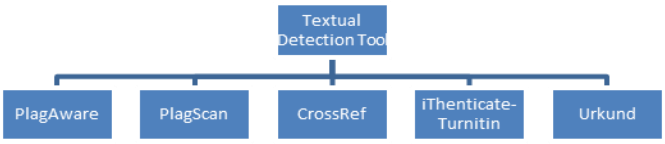

Fig. 2 Textual Detection Tools

We have concentrated in textual plagiarism detection tools and discussed how to improve their performance using NLP techniques. Internally all the tools use SCAM algorithm. SCAM stands for Standard Copy Analysis Mechanism. It detects overlap of two documents using similarity measure and also the difference of size between two documents. Problem of plagiarism detection tools can be categorized into following major parts:

False Positive-When some portion of the text has not been copied at all, the tool is showing as a copy. For example when showing some references and author has quoted some text from other author and also properly acknowledged it the tool counts it as plagued text and include in the percentage.

False Negative-Suppose the author has changed the sequence of words or used synonyms in some specific intervals but in reality the author has stolen the idea or text without mentioning it. The tool fails to detect as SCAM algorithm has no such provision.

Translation- Suppose the author has translated the text from some other sources. Source text is in language A and plagued text is language B. But there is line by line translation without acknowledging the actual author. The plagiarism tools fail to detect it. This is a dangerous problem and in the last ten years as per statistics there are huge Chinese texts which have been copied(translated) from some other languages of the world without any real contribution.

IV. PROPOSED WORK

Semantic similarity is a specialized domain of NLP where contextual proximity between two words or two sentences or two paragraph is discussed. Semantic proximity can also be used instead of semantic similarity or semantic distance as an opposite concept. Numerical score that quantifies similarity/proximity is used to measure semantic similarity. In existing researches, various semantic similarity measures (SSMs) have been defined and many semantic similarity computational models have been proposed. Semantic similarity measures.SSM have been widely used in many NLP and related fields such as information retrieval, text classification, information extraction, machine translation, word sense disambiguation, question answering, plagiarism detection, etc. Semantic similarity calculation can be over words or sentences or paragraphs or even the corpus so it has level of granularity. Regarding semantic similarity the measurement granularity level should cover the full text. Semantic similarity score between suspected document and one or more source documents normally indicates the existence of plagiarism. The whole procedure of semantic similarity measurement and thus plagiarism identification, can be calculated on the sentence and paragraph level. Semantic similarity measures or SSM is used in NLP for various tasks. In 1970 first use of SSM is found for information retrieval. There are several approaches in detecting semantic similarity. Corpora-based and knowledge-based are the two main approaches in measuring the semantic similarity of texts. Ontology-based semantic similarity measures are also extensively described. These measures are basically class of knowledge-based measures. Here we have presented another approach in which we classify measures it two categories:

SSMs on the document/text level and SSMs on the concept/word level.

A. Semantic Similarity on the document/text level

In the document level similarity measure was originally proposed by Salton et al. for the information retrieval related work. He proposed Vector Space Model (VSM) for measuring semantic similarity between two or more documents. After these other researchers also proposed other ways or models mostly all are in the machine learning domain. We consider each document as a point in n-dimensional space in VSM model. Considering a given set of l documents D = $\{D_1,D_2,\ldots D_l\}$, a document D_i is represented as a vector = $(W_{i1},W_{i2}...W_{in})$. Where W_i is the word present in the document D . As per classical approach of the VSM, each of the dimension corresponds to one word or term in the document set of D. Weights is generally assigned by various weighting schemes; TF-IDF is one of the frequently used approach. The similarity between two documents D_i and D_j thus calculated if as cosine similarity can be expressed mathematically :

$$sim_{cos} = \frac{D_i . D_j}{||D_i|| . ||D_j||} \quad (5)$$

High dimensionality, sparseness are the main drawbacks of this VSM model. Additionally uncommon words or vocabulary problems also gives a set back during vector calculation. Therefore, many scientists have worked to improve this classical VSM. Most prominent approaches are Latent semantic analysis (LSA) proposed by Landauer (1998, LSA),Salient Semantic Analysis (SSA), Explicit Semantic Analysis (ESA), Distributional Similarity, Hyperspace Analogues to Language, etc.

B. Semantic Similarity on the concept/word level

On the word level similarity is measured between words. Here input is a pair of words. System uses taxonomy, ontology, wordnet and returns a numerical value based on the semantic similarity of words. Hierarchy is considered also during calculation of the numerical value. Gloss in the wordnet also taken into account for calculation of similarity. From word similarities using different equations we can get similarity between text also. If we have an ontology or "IS A" relationship between two words or concepts then their similarity can be measured by counting the number of edges (Edge based approach) or number of nodes (node based approach) between the two words placed in the ontological tree. In both the approach lower distance means lower numerical value thus more similar. The easiest way to measure similarity of two concepts or words c1 and c2 is to estimate the distance between them alongside the shortest-path joining them. The sophisticated approaches assign variable weight to the edges joining two nodes and taking the weighted average. In most cases, the semantic similarity of two word or concepts is measured as a function of the depth of the Least Common Subsumer (LCS) or Least Common Ancestor (LCA). An example of this approach is Wu and Palmer metrics which defines mathematically as:

$$sim_{wp}(c_1,c_2) = \frac{2depth(LCA_{c1,c2})}{depth(c_1).depth(c_2)} \quad (1)$$

There are two ways in the node based approach 1. Feature based approach 2.based on information theory. In feature based approach a set of features F is used to describe a concept. Now these F features can be compared using classical binary theory or distance measure. An example of this concept Match similarity measure is defined by Maedche and Staab.It's mathematical expression is:

$$sim_{cMatch}(c_1,c_2) = \frac{|F(c_1) \cap F(c_2)|}{|F(c_1) \cup F(c_2)|} \quad (2)$$

Shanon's theory is used for the concept based on information theory. The amount of common information or data is the measurement for similarity. The mathematical expression defined by Resnik by this approach is:

$$sim_{Resnik}(c1,c2) = \max_{c \in s(c1,c2)} [-\log_p(c)] \quad (3)$$

Where $s(c_1,c_2)$ is the set of concepts that subsume both concepts C_l and C_2.

Turney proposed Point wise Mutual Information(PMI). No. of word co-occurrence counts collected over very large corpora is used to calculate PMI. For two words W1 and W2, their PMI-IR is measured as:

$$sim_{PM1}(w_1,w_2) = \log_2[p(w_1,w_2)/(p(w_1).p(w_2))] \quad (4)$$

There are many other concept level measures, named by their authors, amongst whom the important ones are: Leacock & Chodorow,Wu and Palmer, Jiang and Conrath, Resnik Lin, Zhong, Nguyen and AI-Mubaid, Caviedes and Cimino, Lesk etc.

C. Our Approach : Hybrid Method

We propose to use first document level semantic similarity measure and thereby finding the suspicious sources and then to use word level semantic similarity measure to find out the exact sentences. Combination of different algorithms in both the levels is our main target to improve the result accuracy.

V. CONCLUSION and Future Scope

In this paper we have discussed various kinds of plagiarism detection methods and tried to show how plagiarism detection can improve with the semantic analysis and other NLP techniques with the use of some external knowledge base. External knowledge can be represented in ontology or

simpler taxonomies such as WordNet. However, the formalism is not limited to these classical ontologies or taxonomies; it can be any kind of graph representation of lexical relations Another approach is to use statistical methods designed in the domain of NLP that have been used recently for plagiarism detection.

The drawback of the approach is based solely on the semantic similarity measures which are not enough and that they can be combined with classical approaches that may identity copy-paste plagiarism. For further research to experiment with the NOK method or some other graph based formalism for lexical relation representation is suggested. The searchers are experimenting ontology-based information retrieval in which the classical VSM is projected onto a smaller vector space.

SCAM algorithm can be modified for distributed computing platform using some NoSql or Bigdata database. By this we can reduce the execution time enhances the parallelism that will improve the response time.

REFERENCES

[1] Tsatsaronis, George, et al. "Identifying free text plagiarism based on semantic similarity." Proceedings of the 4th International Plagiarism Conference. 2010.

[2] S. Fernando and M. Stevenson, "A semantic similarity approach to paraphrase detection", Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics, 2008.

[3] Shenoy, Manjula K., K. C. Shet, and U. Dinesh Acharya. "Semantic plagiarism detection system using ontology mapping." Advanced Computing 3.3 2012, pp 59.

[4] Le Huong T., et al. "Semantic text alignment based on topic modeling." Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2016 IEEE

[5] Alzahrani, Salha and Naomie Salim. "Fuzzy semantic-based string similarity for extrinsic plagiarism detection." Braschler and Harman 1176 , 2010, pp 1-8

[6] Marsi, Erwin, and Emiel Krahmer. "Construction of an aligned monolingual treebank for studying semantic similarity." Language resources and evaluation 48.2, 2014, pp 279-306.

[7] Vrbanec, Tedo, and Ana Meštrović. "The struggle with academic plagiarism: Approaches based on semantic similarity." The 40th Jubilee International ICT Convention–MIPRO 2017.

[8] Zu Eissen, Sven Meyer, and Benno Stein. "Intrinsic plagiarism detection." European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2006.

[9] Stein, Benno, and Sven Meyer Zu Eissen. "Near similarity search and plagiarism analysis from data and information analysis to knowledge Engineering". Springer, Berlin, Heidelberg, 2006, pp 430-437.

[10] Meuschke, Norman, and Bela Gipp. "State-of-the-art in detecting academic plagiarism." International Journal for Educational Integrity 9.1, 2013.

[11] Chong, Miranda, Lucia Specia, and Ruslan Mitkov. "Using natural language processing for automatic detection of plagiarism." Proceedings of the 4th International Plagiarism Conference, 2010.

[12] Gharavi, Erfaneh, et al. "A Deep Learning Approach to Persian Plagiarism Detection." FIRE (Working Notes), 2016.

[13] Agirre, Eneko, et al. "Semantic textual similarity, monolingual and cross-lingual evaluation." Proceedings of the 10th International Workshop on Semantic Evaluation, 2016.

[14] Kong, Leilei, et al. "Detecting High Obfuscation Plagiarism: Exploring Multi-Features Fusion via Machine Learning." International Journal of u-and e-Service, Science and Technology, 2014, pp 385-396.

[15] A Das and D Saha, "Improvement of Electronic Governance and Mobile Governance in Multilingual Countries with Digital Etymology Using Sanskrit Grammar," in *Social Transformation – Digital Way. CSI 2018*, 2018, pp. 523 - 530.

[16] Arijit Das, Tapas Halder, and Diganta Saha, "Automatic extraction of Bengali root verbs using Paninian grammar," *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 953 - 956.

[17] Arijit Das and Diganta Saha, "Improvement of electronic governance and mobile governance in multilingual countries with digital etymology using sanskrit grammar," *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 502 - 505.

Authors Profile

Mr. Arijit Das pursed Bachelor of Technology in Computer Science and Engineering from 2007 to 2011 and Master of Engineering in Computer Science and Engineering from Jadavpur University in the year 2011-2013 with GATE scholarship. He is currently pursuing Ph.D.(Engg.) from the department of CSE,Jadavpur University and currently working as Assistant Professor in Department of CSE&IT,JUIT,Waknaghat,Solan,Himachal Pradesh,India. He is a member of IEEE(USA) & IEEE(India) since 2016, a life member of the Computer Society of India since 2016, ACM since 2015. He has published more than 5 research papers in reputed international conferences, journals and as book chapter including IEEE,springer nature and Consortium of Indonesian Journals published by Govt. of Indonesia and it's also available online. His main research work focuses on Semantic Searching, NLP, Text Mining, Information Retrieval,Big Data Analytics, Data Mining.He has 3 years of teaching experience and 4 years of Research Experience and 4 years of Industrial Experience.

Ms Shipra Shaw pursued B.Tech. in Computer Science and Engineering from MAKAUT,West Bengal,India. She is currently pursuing Master of Engineering in Computer Science and Engineering from Jadavpur University with GATE scholarship.She has research interest in semantic similarity and NLP.

