Cross lingual Text re-use Detection using Information Retrieval

Aniruddha Ghosh, Santanu Pal, Sivaji Bandyopadhyay

Department of Computer Science and Engineering,
Jadavpur University, Kolkata – 700032, India
{ arghyaonline, santanu.pal.ju }@gmail.com, sivaji_cse_ju@yahoo.com

Abstract. This paper reports about the development of a cross-language text re-use detection system as a part of the cross-language text re-use detection task in FIRE 2011. Here the cross-language text re-use detection is treated as a problem of Information Retrieval and it is solved with the help of Nutch, an open source Information Retrieval (IR) system. Our system contains three phases – knowledge preparation, candidate retrieval and cross-language text reuse detection. In knowledge preparation, stems, hyponyms, hypernyms and synsets of each word are considered which are from WordNet 3.0. which are used to identify the re-used words i.e. the words that are similar in sense to the original words. Knowledge files are indexed using Lucene. Suspicious documents are in Hindi. Hence, each suspicious document is translated using Google Hindi-English translator. Nutch performs a paragraph-paragraph mapping for a proximal match between the suspicious documents and indexed source files. From the probable set of matched source documents, top 5 source documents are chosen as text-reuse based on frequency. The candidate source documents are again pruned using a Dissimilarity score between each set of candidate documents and the suspicious document. The cross-language text re-use detection system was evaluated using the evaluation framework and posed F-score of 0.220.

Keywords: Nutch, Text-reuse, Information Retrieval, Search Engine.

1 Introduction

Plagiarism or Text-reuse is defined as close imitation or purloining and publication of another author's language, thoughts, ideas, or expressions, and the representation of them as one's own original work. From 18th century, plagiarism has been considered as academic dishonesty (Wikipedia). For decades, researchers have explored different techniques to detect plagiarism. In our approach, we have considered cross lingual text-reuse detection problem as an Information Retrieval (IR) problem. For text-reuse detection, our developed system consists of three phrases – knowledge preparation, suspicious paragraph and probable set of source paragraph pair and finally text-reuse detection for each of the suspicious documents.

We have used our previous developed system for plagiarism or text-reuse detection (Ghosh et. al., 2011). To perform paragraph level text-reuse detection, the algorithm has been modified. A Google translation tool is used to translate suspicious Hindi documents into English.

Due to absence of controlled evaluation environment to compare results of the algorithms, text-reuse detection is still a challenging task (Potthast, et al., 2009). The re-use detection task becomes more complex, especially if language of suspicious document is Indian Language as it is morphologically rich in construction. Researchers have organized various conferences to overcome text re-use problem. Among the attempts made by researcher, fingerprint retrieval method (Palkovskii et. al., 2010), candidate retrieval (Pereira et. al., 2010), passage retrieval (Vania et. al. 2010) are most prominent. Mozgoyov et. al., 2007 has developed natural language parser to find swapped words and phrases to detect intentional plagiarism while n-gram co-occurrence statistic to detect verbatim copy while Longest Common Subsequence is used to handle text modification by Chen et.al., 2010. Researchers have used cosine similarity score and n-gram vector space model at different levels i.e. word (Grozea et. al., 2009) and character (Basile et al., 2009).

Here we have treated text-reuse as IR problem. Hence we have used Nutch, an open source search engine, to retrieve the re-used portion from the suspicious documents. The following paper is organized as follows: section 2 gives the statistics about the test corpus. Section 3 describes the system framework. Section 4 describes Knowledge preparation, Section 5 describes candidate retrieval and Section 6 describes post processing. Then in Section 7 evaluation scores are given and Section 8 concludes this work and shows the future direction.

2 Corpus Statistics

The data set consists of 5032 source documents and 190 suspicious documents. The source documents are in English and suspicious documents are in Hindi. Source documents are wiki files.

3 System Framework

In this section, we describe our Information Retrieval (IR) based Text-reuse Detection system framework as shown in the figure 1. The system is defined in three parts: Knowledge Preparation, suspicious paragraph and probable set of source paragraphs pair identification and finally test-reuse detection of each suspicious document.

The Nutch[1] IR system has been used for the present task. Nutch follows the standard IR model of Lucene[2] with Document parsing, Document Indexing, TF-IDF calculation, query parsing and finally searching/document retrieval and retrieved

[1] http://nutch.apache.org/
[2] http://lucene.apache.org/

documents ranking. Some modules in Nutch have been upgraded for our present need as described below.

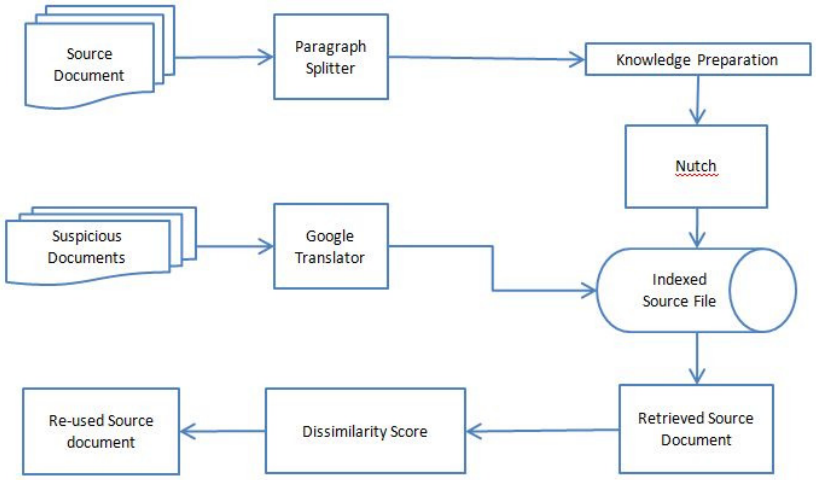

Fig. 1. System Architecture

4 Knowledge Preparation

After deep analysis of the data set, it is observed that test-reuse extraction at paragraph level is desired.

4.1 Source Document Parsing

The source documents are wiki files. Hence, the source documents are cleaned to extract the raw text. In Source document Parsing module, each source document has been split into files paragraph wise i.e. each files contains one paragraph.

4.2 Knowledge File Generation

After dividing the source documents into paragraphs, knowledge files are generated for each of the paragraphs. Initially each knowledge file contains a single source sentence. The file names of knowledge files are created in such a manner that the source paragraph in the original source document can be tracked. E.g. name of the knowledge file with the 100th source paragraph of source document named 'source-document00002.txt' is "source-document00002.100.txt"

4.3 Knowledge Build

To build the knowledge of the paragraph in the knowledge file, the stem, synonyms, hyponyms, hypernyms and synsets of each word except stop words are extracted from WordNet 3.0[3] and duplicate words are removed to get the set of similar sensed unique words. Now all the similar senses are added along with the original word in the knowledge file.

The stem form of a word was considered for checking the text te-use between two inflected lexical forms of the same word. The synonym like semantic property helps to identify whether two words are plagiarized or not. Hypernym is the semantic relation in which words stand when their expansions stand in the relation of class to subclass. E.g. if a source document contains a word 'dog', but the author of a the suspicious document refers the same information as 'animal', then hypernym relation between 'animal' and 'dog' will help to identify the text-reuse. Similarly we check their hyponymy relation between words in source and suspicious paragraphs to identify reverse semantic relation. The synset has given the set of similar senses of a word, which has identified the similar sensed word between the source and suspicious sentences, those words has considered as plagiarized words.

4.4 Source Document Indexing

After building the knowledge files, they are indexed using Lucene, an open source full text search. The basic architecture of Lucene is shown in figure 2.

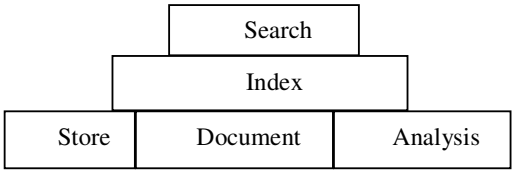

Fig. 2. Lucene Architecture

5 Candidates Retrieval

The suspicious documents are in Hindi. Hence, we have used Google translation system to translate the files to English. To retrieve the candidate set of each sentence of suspicious documents, the following method is used.

[3] http://wordnet.princeton.edu/

5.1. Suspicious Document Parsing

In Suspicious document Parsing module, each suspicious document has been parsed to identify and extract all sentences in the suspicious documents. Each Hindi sentence is translated using Google translator to English. The translated sentences are used as query to Nutch.

5.2 Source Candidate Retrieval

After generating the query from the suspicious sentences, they have been fired to the Information Retrieval system, Nutch to retrieve the probable set of source sentences corresponding to each suspicious sentence. As source documents are split into paragraphs into files and each file contains only one paragraph, Nutch performs a sentence-paragraph mapping for a proximal match between the query and indexed source files with single paragraph. Nutch retrieves among the indexed files the set of probable source files if matched. After the candidate retrieval, Nutch gives a set of probable candidates in ranked order with their similarity scores for each suspicious sentence, which were matched with at least one source sentence. The retrieved documents are collected and stored in a file for all the sentences of a certain paragraph of suspicious documents. From the retrieved documents we have trimmed the paragraph identification mark i.e. "source_0002.3.txt" is replaced to "source_0002.txt". Afterwards, frequency of each of the retrieved source documents is calculated. The highest ranked source document has the highest proximal match with suspicious document paragraph. The top 5 ranked documents are taken into consideration. For each paragraph of the suspicious documents, five source documents are detected. Afterwards, all the retrieved source documents for each paragraph of a suspicious document is grouped together to perform a document level text-reuse detection.

6 Text-reuse Detection

After examining the retrieved source documents for a certain suspicious document, we found that match between retrieved source documents and suspicious document can vary from a single word to whole paragraph. But text-reuse is considered if similarity is found by a big amount. Hence, A Text-reuse Detection module pruning method is implemented among the retrieved source documents with the help of Dissimilarity Measurement score (Keseljy et. al., 2003).

6.1 Dissimilarity Measurement

The algorithm 1 gives the algorithm for calculating the dissimilarity between the suspicious document and its corresponding retrieved candidate documents. Between two documents that have most identical n-grams, the dissimilarity score is almost 0. Consider two documents as s1 and s2. s1 document contains the unigrams of a certain

suspicious document whereas s2 contains the unigrams of a retrieved candidate document. Given two documents the algorithm returns a positive number, which is a measure of dissimilarity.

Algorithm 1: Document Dissimilarity (Document1, Document2)
1: sum= 0 2: for all n-grams x contained in Document1 or Document2 do 3: let f1 and f2 be frequencies of x in Document 1 and Document 2 (zero if they are not included) 4: add square of the normalized difference of f1 and f2 to sum: sum=sum + $(2*(f1- f2)/(f1 + f2))^2$ 5: return sum

For each of the retrieved source document, dissimilarity score is calculated. We can interpret the dissimilarity score in an absolute way, i.e., an optimal threshold dissimilarity score value has been determined through a brief overview over the scores of retrieved source documents for a certain suspicious document. Every document whose score is below the threshold value is considered as text-reuse.

7 Evaluation

Based on the evaluation measure given by the organizer, our system has posed F-score . The detail score of our system can be seen in Table 1.

Table 1. Evaluation

Measurement	Score
Precision	0.226
Recall	0.214
F-score	0.220

8 Conclusion and Future Works

Our system has posed F-score .220. The source files are wiki files which contains a huge noise in the file. Noises are removed as much as possible. Threshold value of dissimilarity score is chosen quite low which has chosen almost exact text-reuse between the source document and suspicious document. Hence recall value is quite low. Experimentation on more relaxed threshold value is still due.

We have used Google translator for machine translation from Hindi to English. Improvement of machine translation will cast significant change over accuracy. For the building up the knowledge base, we have used the surface word and its' hypernym and synsets. We will try to use the root forms of the words and their synset and

hypernym to build the knowledge base. All these experiments will enhance the result to a great extend.

References

1. Viviane P. Moreira Rafael C. Pereira and Renata Galante. UFRGSPAN2010: Detecting External Plagiarism: Lab Report for Pan at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.
2. Yurii Palkovskii, Alexei Belov, and Irina Muzika. Exploring Fingerprinting as External Plagiarism Detection Method: Lab Report for PAN at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.
3. Clara Vania and Mirna Adriani. External Plagiarism Detection Using Passage Similarities: Lab Report for PAN at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.
4. Cristian Grozea and Marius Popescu. Encoplot—Performance in the Second International Plagiarism Detection Challenge: Lab Report for PAN at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.
5. M. Mozgovoy, T. Kakkonen, and E. Sutinen. Using Natural Language Parsers in Plagiarism Detection. In Proceeding of SLaTE'07 Workshop, Pennsylvania, USA, October 2007.
6. Basile et al. 2009. A Plagiarism Detection Procedure in Three Steps: Selection, Matches and "Squares". In Stein et al. (Stein et al., 2009).
7. Potthast, Martin et al. 2010. An Evaluation Framework for Plagiarism Detection. In Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China, August 2010. Association for Computational Linguistics.
8. Clara Vania and Mirna Adriani. 2010. Automatic External Plagiarism Detection Using Passage Similarities. In the Proceedings of the Uncovering Plagiarism, Authorship, and Social Software Misuse, PAN 2010, CLEF 2010.
9. en.wikipedia.org/wiki/Plagiarism
10. Aniruddha Ghosh, Pinaki Bhaskar, Santanu Pal, Sivaji Bandyopadhyay. Rule Based Plagiarism Detection using Information Retrieval. Notebook for PAN at CLEF 2011.
11. Vlado Keseljy, Fuchun Pengz, Nick Cerconey, Calvin Thomasy. N-gram-based author profiles for authorship Attribution. Pacific Association for Computational Linguistics, 2003.