# BAENPD: A Bilingual Plagiarism Detector

Mohammad Shamsul Arefin†, Yasuhiko Morimoto†, Mohammad Amir Sharif‡
†Graduate School of Engineering, Hiroshima University, Japan
‡The Center for Advanced Computer Studies, University of Louisiana at Lafayette, USA
Email: d105660@hiroshima-u.ac.jp, morimoto@mis.hiroshima-u.ac.jp, mas4108@cacs.louisiana.edu

*Abstract*— **In this paper, for the first time, we present a bilingual plagiarism detection system, BAENPD that can detect plagiarism from electronic Bangla and English documents. It uses two different methods for detecting plagiarism. The first method is based on the analysis of individual contents of the documents, whereas second technique performs several statistical analysis of the documents. The system has been evaluated by real documents. We have found that our system can efficiently detect plagiarism between English and Bangla documents as well as from the documents of same language.**

*Index Terms*— **Plagiarism, statistical analysis, query execution, documents relevancy, root detection.**

## I. INTRODUCTION

Plagiarism is defined as the appropriation or imitation of the language, ideas and thoughts of another author and representation of them as one's original work. It means using another author's work without giving him proper credit. Though plagiarism can be found in almost every field, it is a major problem in academic areas as plagiarism destroys individual's creativity and originality and defeats the purpose of education. For recognizing creativity and originality of one's work, it is necessary to detect plagiarized contents efficiently.

The easy availability of information in present days makes it easier to plagiarize another author's contents without proper citation or reference and this tendency is increasing day-by-day. Although there are many commercial and non-commercial tools available for plagiarism detection, most of them are uni-lingual in nature and none of them can detect plagiarism in Bangla documents. However, at present in academic and non-academic areas many works are carried out in Bangla and we need to be aware of the originality of these works. An efficient plagiarism detector considering the information in Bangla can ensure the novelty of such works.

Considering the above facts, we have proposed two different methods for detecting plagiarized contents between English and Bangla documents as well as from the documents of the same language. Our first approach is based on the overall contents of the documents and is well applicable in detecting plagiarism from documents of any size and any domain. However, our second methodology depends on the statistical information of the documents and it performs better if the documents are from the same domain and uniform in size.

## II. MOTIVATION

Plagiarism from the contents of a language to another language is a common problem nowadays. This type of plagiarism often occurs when the target language has less resource like Bangla language. Consider the representation of short information about the St. Martin's Island of Bangladesh in English and Bangla, as shown in Figure 1 and Figure 2.

Let us first consider the information in Figure 1. From Figure 1, we can clearly see that though the information is in two different languages, the sentence structures and order of sentences are almost same. Also, similar set of keywords i.e. keywords in a language and the corresponding translated words in another language, yields the same content, even though keywords are placed in different order for keeping the accurate structure of the sentences in each language.

Then, looking at the information of Figure 2, we observe that though the information in English is same as the information in Figure 1, the information in Bangla is different than the information in Figure 1. It is because the sentences in Bangla has been created from source information ( here from English) by reordering the sentences and using different words than the words use in Figure 1.

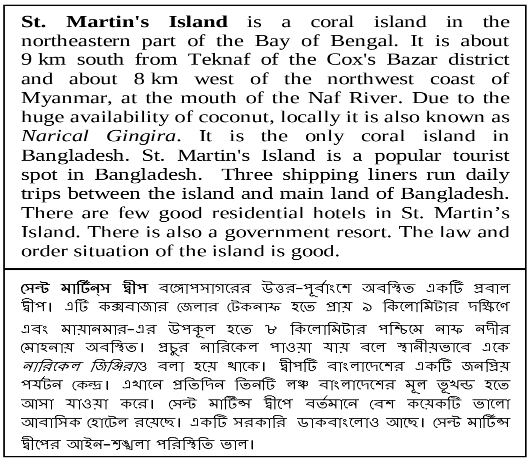
However, if we look carefully in the information of both figures, we can find that in both cases information in Bangla has been plagiarized from the same English document. The information in Bangla in Figure 1 has been almost directly plagiarized whereas in Figure 2 the information has been plagiarized using some level of intelligence. The second type of plagiarism is the most common form of plagiarism as identification of such type of plagiarism is often hard to detect. The second form of plagiarism is also common in the documents of the same language.

Considering the above facts, we have developed two plagiarism detection techniques those can efficiently detect plagiarism from the documents of two different languages English and Bangla as well as from the documents of the same language domain.

The contributions of this paper can be summarized as follows. First, we propose efficient techniques for identifying stop words and synonyms. Second, we elaborate

This paper is based on "Bilingual Plagiarism Detector," by Mohammad Shamsul Arefin, Yasuhiko Morimoto, and Mohammad Amir Sharif, which appeared in the Proceedings of the 14th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, December 22-24, 2011. This paper in an enhanced version with detail explanations and additional concepts.

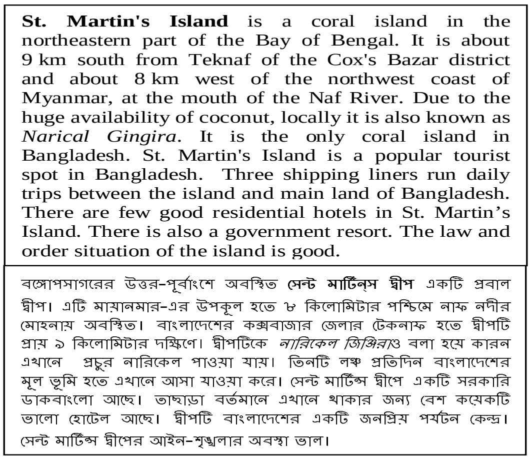
doi:10.4304/jcp.8.5.1145-1156



**St. Martin's Island** is a coral island in the northeastern part of the Bay of Bengal. It is about 9 km south from Teknaf of the Cox's Bazar district and about 8 km west of the northwest coast of Myanmar, at the mouth of the Naf River. Due to the huge availability of coconut, locally it is also known as *Narical Gingira*. It is the only coral island in Bangladesh. St. Martin's Island is a popular tourist spot in Bangladesh. Three shipping liners run daily trips between the island and main land of Bangladesh. There are few good residential hotels in St. Martin’s Island. There is also a government resort. The law and order situation of the island is good.

**সেন্ট মার্টিন্‌স দ্বীপ** বঙ্গোপসাগরের উত্তর-পূর্বাংশে অবস্থিত একটি প্রবাল দ্বীপ। এটি কক্সবাজার জেলার টেকনাফ হতে প্রায় ৯ কিলোমিটার দক্ষিণে এবং মায়ানমার-এর উপকূল হতে ৮ কিলোমিটার পশ্চিমে নাফ নদীর মোহনায় অবস্থিত। প্রচুর নারিকেল পাওয়া যায় বলে স্থানীয়ভাবে একে *নারিকেল জিঞ্জিরা*ও বলা হয়ে থাকে। দ্বীপটি বাংলাদেশের একটি জনপ্রিয় পর্যটন কেন্দ্র। এখানে প্রতিদিন তিনটি লঞ্চ বাংলাদেশের মূল ভূখন্ড হতে আসা যাওয়া করে। সেন্ট মার্টিন্স দ্বীপে বর্তমানে বেশ কয়েকটি ভালো আবাসিক হোটেল রয়েছে। একটি সরকারি ডাকবাংলোও আছে। সেন্ট মার্টিন্স দ্বীপের আইন-শৃঙ্খলা পরিস্থিতি ভাল।

Figure 1. Example of direct plagiarism



**St. Martin's Island** is a coral island in the northeastern part of the Bay of Bengal. It is about 9 km south from Teknaf of the Cox's Bazar district and about 8 km west of the northwest coast of Myanmar, at the mouth of the Naf River. Due to the huge availability of coconut, locally it is also known as *Narical Gingira*. It is the only coral island in Bangladesh. St. Martin's Island is a popular tourist spot in Bangladesh. Three shipping liners run daily trips between the island and main land of Bangladesh. There are few good residential hotels in St. Martin’s Island. There is also a government resort. The law and order situation of the island is good.

বঙ্গোপসাগরের উত্তর-পূর্বাংশে অবস্থিত **সেন্ট মার্টিন্‌স দ্বীপ** একটি প্রবাল দ্বীপ। এটি মায়ানমার-এর উপকূল হতে ৮ কিলোমিটার পশ্চিমে নাফ নদীর মোহনায় অবস্থিত। বাংলাদেশের কক্সবাজার জেলার টেকনাফ হতে দ্বীপটি প্রায় ৯ কিলোমিটার দক্ষিণে। দ্বীপটিকে *নারিকেল জিঞ্জিরা*ও বলা হয়ে কারন এখানে প্রচুর নারিকেল পাওয়া যায়। তিনটি লঞ্চ প্রতিদিন বাংলাদেশের মূল ভূমি হতে এখানে আসা যাওয়া করে। সেন্ট মার্টিন্স দ্বীপে একটি সরকারি ডাকবাংলো আছে। তাছাড়া বর্তমানে এখানে থাকার জন্য বেশ কয়েকটি ভালো হোটেল আছে। দ্বীপটি বাংলাদেশের একটি জনপ্রিয় পর্যটন কেন্দ্র। সেন্ট মার্টিন্স দ্বীপের আইন-শৃঙ্খলার অবস্থা ভাল।

Figure 2. Example of indirect plagiarism

the method for detecting root words that improves the efficiency of plagiarism detection. Third, we provide a technique for detecting plagiarism based on synonyms and keywords in the documents. Fourth, we perform statistical analysis of the documents and based on statistical results, we detect plagiarism in the documents. Fifth, we give the complexity analysis of the proposed methods. Finally, we conduct extensive experiments on a set of real documents and show the efficiency and robustness of our proposals.

The remainder of the paper is organized as follows. Section III provides a brief review of related works on plagiarism detection. Section IV describes the preliminary ideas about different related topics. Section V details architecture and methodology of content based plagiarism detection technique. In Sections VI, we present statistical approach of plagiarism detection. Section VII gives the experimental results. Finally, we conclude the paper in Section VIII.

## III. Literature Review

In this section, we are providing a brief overview of the initiatives taken around the globe for detecting plagiarism. WriteCheck [1] plagiarism detector uses pattern recognition to match the contents of submitted papers against Internet resources and then against an in-house database of previously submitted papers. This technique is different than the text searches of popular search engines such as Google and Bing and produces less false positives than search technology designed for other purposes. EVE2 [2] is a windows based system that performs a reliable check of similar contents from the Internet to track down possible instances of plagiarism. It examines the essays and then quickly makes a large number of complex searches in the Internet to locate suspected sites. DOC Cop [3] is a plagiarism, cryptomnesia and collusion detection tool that creates reports displaying the correlation and matches between documents or a document and the Web. Plagium [4] is a fast and easy-to-use tool to check text against possible plagiarism or possible sources of origination. Plagium intelligently breaks up the input text into smaller snippets. These snippets are matched against web content in an efficient manner. The matches scores are used to determine the matching of documents with the input text.

Plagiarism Detector [5] is a software tool to effectively discover, trace and prevent unauthorized copy-pasting of any textual material taken from the world wide web. It does not use any in-house database. Rather, it uses the databases of three well known search engines: Google, Yahoo, and AltaVista. Upon receiving a document, at first it splits the document into different phrases and each phrase is sent to three different search engines. Next, the result of every search engine is downloaded and result sources are analysed. Finally, merges the analysed reports and generates the originality report of the submitted document. CodeMatch [6] compares thousands of source code files in multiple directories and subdirectories to determine which files are the most highly correlated. It first divides each source code into elements and then determines the correlation using several matching algorithms. This can be used to significantly speed up the work of finding source code plagiarism. CodeMatch is also useful for finding open source code within proprietary code, determining common authorship of two different programs, and discovering common, standard algorithms within different programs. Pl@giarism [7] is a plagiarism detection tool developed at the Law Faculty of the University of Maastricht, Belgium. Pl@giarism performs cross-comparison against each other in order to detect similarities among the documents in the sample. It determines similarities between pairs of documents by comparing three word phrases in each. Main limitation of Pl@giarism is that it does not automatically check against sources on the Internet. GPSD [8] is a screening program that has been designed to help users become more sensitive to their own writing style. GPSD just provides a rough estimate whether plagiarism has or has not occurred. GPSP [9] evaluates a student’s knowledge of their own writing by producing a test. Every fifth word of a student’s paper is eliminated and replaced with blanks which the student has to replace. Accuracy and speed in replacing the blanks is evaluated against a proprietary database, and a probability score returned immediately.

It is useful for detecting plagiarism where the original source cannot be located through other sources such as Internet search engines and other plagiarism detection services. Its limitations include not being able to identify the source of the suspect text and requirement for students to sit a test. WCopyfind [10] software is a delightful example of the power of the computer to help in addressing the plagiarism problem. It makes text-string comparisons and can be instructed to find sub-string matches of given length and similarity characteristics. Such fine tuning permits the exclusion of obvious non-plagiarism cases despite text-string matches. The main limitation of the above works is that they do not consider the detection of plagiarism when the content is plagiarized from a specific language and use in a different language.

Due to the rapid growth of Internet, this fact becomes a major concern among the research communities. MLPlag [11] is an approach of plagiarism detection in multilingual environment. This method is based on analysis of word positions. It utilizes the EuroWordNet thesaurus that transforms words into language independent form. This allows identifying documents plagiarized from sources written in other languages. Special techniques, such as semantic-based word normalization, were incorporated to refine the method. In [12], the authors propose a method for cross-language plagiarism detection. In this work, the authors introduce three models for the assessment of cross-language similarity and perform the comparison among these three models. Cedeno et al. [13] introduce a model for detecting plagiarism across distant language pairs based on machine translation and monolingual similarity analysis.

Though, there are many plagiarism detection tools in commercial and academic areas, there is no model that can check the level of plagiarism of information from other language documents to Bangla documents and vice versa. Even there is no tool that can detect plagiarism among Bangla documents. Considering these facts, in this paper, we provide a framework for detecting plagiarism among Bangla documents as well as plagiarism of information from English documents to Bangla documents and vice versa. Our developed framework can also detect plagiarism among English documents.

## IV. Preliminaries

### A. Plagiarism Detection Systems and Related Issues

The term plagiarism is well known among the research community. It generally focuses on detecting plagiarism from structured documents and considers that there is a large number of documents. The process of plagiarism detection consists of locating relevant documents and plagiarised parts from the documents, on the basis of query documents.

The amount of information available in electronic form is growing exponentially, making it increasingly important to find the plagiarised documents so that the tendency of copying one's information is reduced significantly. In general, a plagiarism detection tool can be used for detecting plagiarism in the newly submitted research papers. It can also be used at the universities for checking uniqueness of the works of the students.

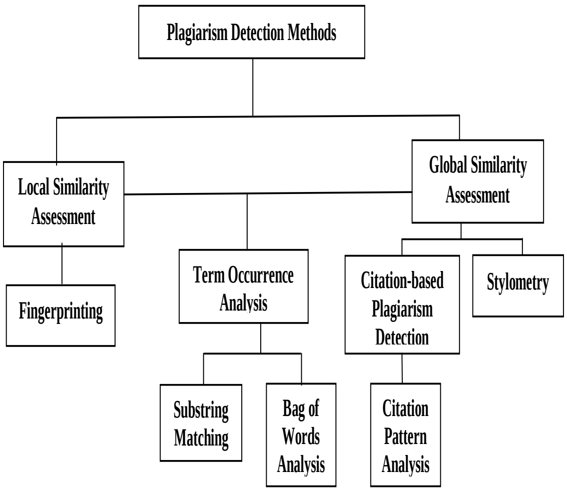

Plagiarism Detection Methods
Local Similarity Assessment
Global Similarity Assessment
Fingerprinting
Term Occurrence Analysis
Citation-based Plagiarism Detection
Stylometry
Substring Matching
Bag of Words Analysis
Citation Pattern Analysis

Figure 3. A taxonomy of plagiarism detection methods [14]

### B. Characteristics of a Bilingual Plagiarism Detector

For developing any bilingual plagiarism detection system, generally we have to consider four main things. Firstly, the detection approach should be in such a way that there is almost no performance degradation of the detector based on the language of the documents. Secondly, there must be an approach for efficient handling of bilingual dictionaries. Thirdly, the document to be checked for plagiarism (query document) need to be processed using similar approach as the approach used for training the detector. Formal way is that if we use algorithm $A$ for training the system, we need to use algorithm $A$ for processing the query document. Lastly, we need a very good framework for proper modelling by which we can find the similarity in the documents. These are the crucial issues in bilingual plagiarism detection.

After getting the relevance, the detector needs to sort the documents in descending order based on the value of the relevance. Then, the top document in the sorted list is marked as most relevant with respect to the query document and so on.

### C. Problem of Bilingual Plagiarism Detection

Finding plagiarism from documents of two different languages is more difficult than identifying plagiarism in the documents of same language. It is even more difficult if the languages are non-ideographic like English and Bangla languages. This is because the structure of documents in non-ideographic languages differs significantly. As for example, there is major difference in the structure of documents in Bangla and English. In Bangla, there are many variations of same words. We can form many different words by just adding some postfixes. But these varied words have similar meaning. On the other hand, in English such variations are less in number.

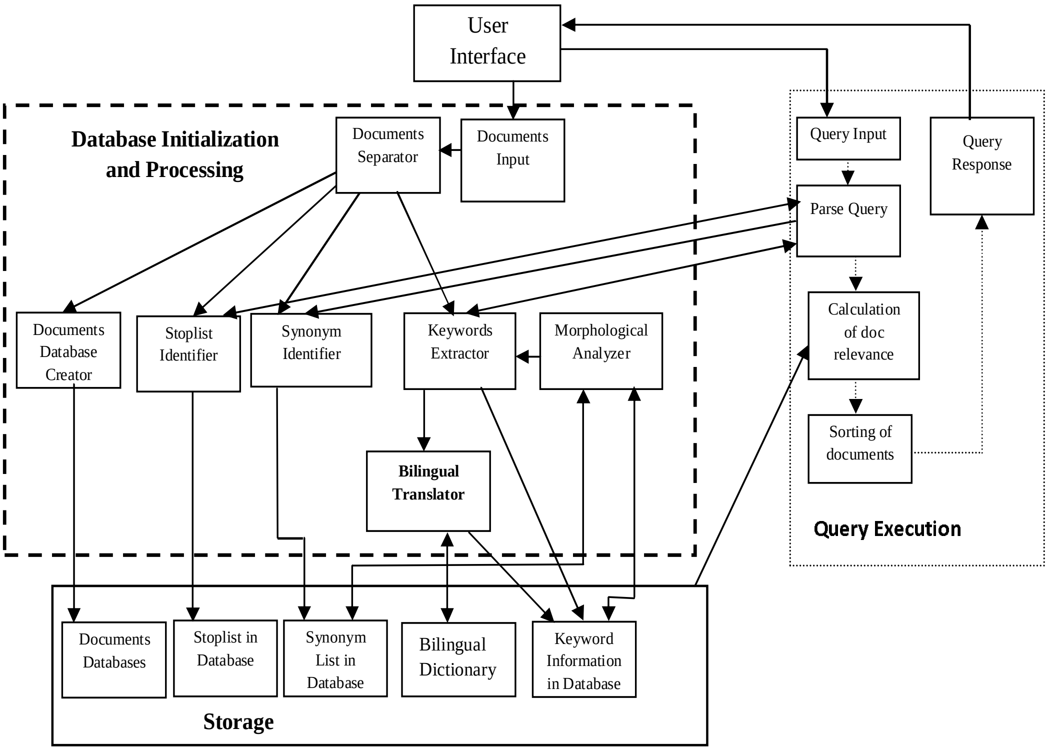


Figure 4. Architecture of content based bilingual plagiarism detection

Also, the use of synonyms in the plagiarized documents and the difference in the sentence structures of the languages makes the detection task harder.

Therefore, we need to develop efficient techniques to solve above problems for appropriate plagiarism detection.

### D. Plagiarism Detection Methods

There are different methods for detecting plagiarism. Figure 3 gives a taxonomy of different plagiarism detection methods. The techniques are characterized by the type of similarity assessment they apply. In global similarity assessments large parts of the text or the document as a whole is used for similarity measure. On the other hand, in local method confined text segments are considered as input.

At present fingerprinting method is most widely used approach in plagiarism detection. The procedure forms representative digests of documents by selecting a set of multiple sub-strings from them. The sets represent the fingerprints and their elements are called minutiae [15], [16]. For checking a suspicious document for plagiarism at first the fingerprint of the document is computed and then perform query minutiae with a pre computed index of fingerprints for all documents of a reference collection.

Substring matching approach [17], [18], [19] uses efficient string matching algorithms. Checking a suspicious document using string matching approach requires the computation and storage of efficiently comparable representations for all reference documents. However, substring matching approach is computationally expensive. So, it is not well applicable for checking large document collections.

Bag of words analysis [20], [21], [22] uses the concept of information retrieval. In this case, documents are represented with one or more vectors. Cosine similarity measure or any other similarity functions can be used for this task.

Citation-based plagiarism detection [23], [24], [25] is based on citation and reference information instead of the text of the documents. This approach can effectively identify similar patterns in the citation sequences of two academic works. In academic areas this method is helpful for checking plagiarism in academic documents.

Stylometry [26] uses statistical methods for identifying individual author's unique writing style and is mainly used for authorship attribution. By constructing and comparing stylometric models for different text segments and passages plagiarization can be detected.

In this paper, we have used two different methods for plagiarism detection. One method is based on the concept of information retrieval. Another method considers statistical information of the documents.

## V. BILINGUAL PLAGIARISM DETECTION BASED ON OVERALL CONTENTS IN THE DOCUMENTS

The system architecture for bilingual plagiarism detection based on overall contents in the documents is given in Figure 4. It comprises three main modules: database initialization and processing module, storage module and query execution module. The database initialization and

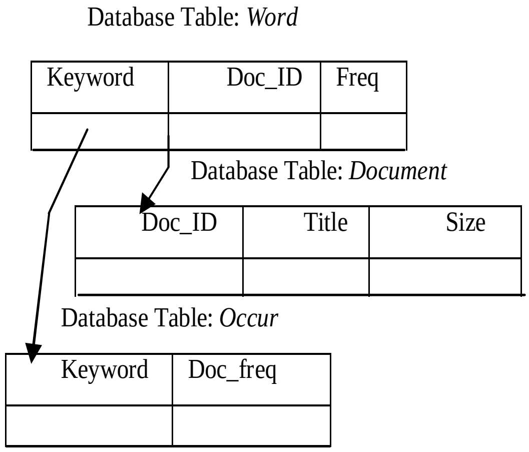
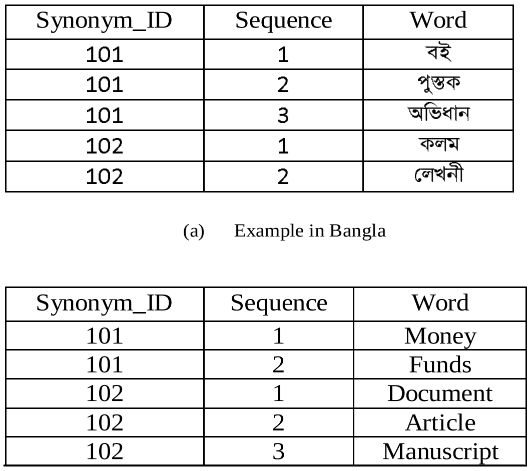


Figure 5. Storage of term information



| Synonym_ID | Sequence | Word |
|---|---|---|
| 101 | 1 | বই |
| 101 | 2 | পুস্তক |
| 101 | 3 | অভিধান |
| 102 | 1 | কলম |
| 102 | 2 | লেখনী |

(a) Example in Bangla

| Synonym_ID | Sequence | Word |
|---|---|---|
| 101 | 1 | Money |
| 101 | 2 | Funds |
| 102 | 1 | Document |
| 102 | 2 | Article |
| 102 | 3 | Manuscript |

(b) Example in English

Figure 6. Synonym information

**Algorithm 1** *Document_DB*

**Input:** Document sets

**Require:** Storing the documents in the file system

```
begin
Create two file objects of the directory name where the document files exists
Create two tables with the field Doc_ID, $Title$, $Size$ having the data type as number, BFILE and number respectively
while counter is not greater than directory length of the directory do
    Insert the file name of directory [counter] into $Title$ field of the corresponding document table.
    Increment the value of counter corresponding to the table by 1
end while
end
```

processing module sets up the database from an initial set of documents against which the query document will be checked for plagiarism. The storage module manages the storage of the text database and related information. Query execution module takes query document as input and checks the relevancy of the document with the documents in the database, sorts the documents in descending order according to their relevance percentage with test documents and returns the result to the user. The single directional arrows represent the direction of next sub-module to be executed in a module and the double directional arrows represent the relationship of a sub-module with sub-modules from which the sub-module gets help for processing.

### *A. Database Initialization and Processing Module*

Database initialization and processing module consists of the sub-modules: documents input, documents separator, documents database creator, stoplist identifier, synonym identifier, keywords extractor, morphological analyzer, and bilingual translator. The relationships among the sub-modules are shown in Figure 4.

*1) Documents Input:* Documents input sub-module takes the set of documents $D = D_{b1}, D_{b2}, \cdots, D_{bn}, D_{e1}, D_{e2}, \cdots, D_{em}$ as input from the user. Here, $D_{bi}, 1 \leq i \leq n$ is the list of documents in one language and $D_{ej}, 1 \leq j \leq m$ is the list of documents in another language. This set of documents is used as training documents. In our proposed method, we consider that the characters in documents are unicode supported.

*2) Documents Separator:* Documents separator sub-module separates the documents set $D$ into two subsets $D_1 = D_{b1}, D_{b2}, \cdots, D_{bn}$ and $D_2 = D_{e1}, D_{e2}, \cdots, D_{em}$ based on the contents of documents. The separation is necessary because the analyzing of documents mainly based on the language of the document's contents.

*3) Documents Database Creator:* Documents database creator module stores each of the document sets $D_1 = D_{b1}, D_{b2}, \cdots, D_{bn}$ and $D_2 = D_{e1}, D_{e2}, \cdots, D_{em}$. These sets of documents are used as training documents. The document databases for both $D_1 = D_{b1}, D_{b2}, \cdots, D_{bn}$ and $D_2 = D_{e1}, D_{e2}, \cdots, D_{em}$ are created with the information document ID, document title and size for each of the documents. In our system, we use BFILE data type to store the document database in a file system. The algorithm for creating and storing the documents in database is given in **Algorithm 1**. **Algorithm 1** stores all the documents in the database as BFILE. The first sql query creates paths from which the BFILE should keep the reference. The second sql command creates two tables declaring document title as BFILE according to the syntax.

*4) Stoplist Identifier:* This sub-module stores some most frequently used terms in the database. These words are auxiliary terms and are used most frequently. These are called stopwords. We call the list of stopwords in our system as stoplist. Stoplist creation module stores the stopwords of each language in a separate database table

| S.N | Postfix | S.N | Postfix | S.N | Postfix | S.N | Postfix |
|---|---|---|---|---|---|---|---|
| 1 | টি | 39 | সমূহ | 77 | ছিলেন | 115 | ইলুম |
| 2 | টির | 40 | গর্ব | 78 | ছিলাম | 116 | ইত |
| 3 | টা | 41 | বর্গ | 79 | ছিলুম | 117 | ইতে |
| 4 | টার | 42 | আবলী | 80 | ছিলেম | 118 | ইতাম |
| 5 | টুকু | 43 | সমুদয় | 81 | ল | 119 | ইতেন |
| 6 | টুকুর | 44 | গুচ্ছ | 82 | লে | 120 | ইতিস |
| 7 | খানা | 45 | গ্রাম | 83 | লি | 121 | ইতেছ |
| 8 | খানার | 46 | বৃন্দ | 84 | লাম | 122 | ইতেছে |
| 9 | খানি | 47 | পুঞ্জ | 85 | লুম | 123 | ইতেছি |
| 10 | খানির | 48 | মণ্ডলী | 86 | লেম | 124 | ইতেছেন |
| 11 | গাছা | 49 | শ্রেণী | 87 | লেন | 125 | ইতেছিস |
| 12 | গাছি | 50 | র | 88 | এ | 126 | ইতেছিল |
| 13 | গুলো | 51 | রা | 89 | এন | 127 | ইতেছিলাম |
| 14 | গুলা | 52 | রে | 90 | এছ | 128 | ইতেছিলে |
| 15 | গুলি | 53 | এর | 91 | এছে | 129 | ইতেছিলি |
| 16 | গুলোর | 54 | এরা | 92 | এছি | 130 | ইতেছিলেন |
| 17 | গুলার | 55 | ব | 93 | এছেন | 131 | ইতেছিলেম |
| 18 | গুলির | 56 | বা | 94 | এছিস | 132 | ইতেছিলুম |
| 19 | দের | 57 | বে | 95 | এছিল | 133 | ইয়াছ |
| 20 | দিগের | 58 | বেন | 96 | এছিলে | 134 | ইয়াছে |
| 21 | গণ | 59 | বি | 97 | এছিলি | 135 | ইয়াছি |
| 22 | জন | 60 | ত | 98 | এছিলেন | 136 | ইয়াছেন |
| 23 | দল | 61 | তে | 99 | এছিলেম | 137 | ইয়াছিস |
| 24 | চয় | 62 | তেম | 100 | এছিলুম | 138 | ইয়াছিলে |
| 25 | সব | 63 | তেন | 101 | এছিলাম | 139 | ইয়াছিলি |
| 26 | মহল | 64 | তাম | 102 | ই | 140 | ইয়াছিল |
| 27 | পটল | 65 | তুম | 103 | ইও | 141 | ইয়াছিলেন |
| 28 | জাল | 66 | তিস | 104 | ইস | 142 | ইয়াছিলেম |
| 29 | দাম | 67 | উন | 105 | ইবে | 143 | ইয়াছিলাম |
| 30 | পাল | 68 | উক | 106 | ইব | | |
| 31 | সকল | 69 | ছ | 107 | ইবি | | |
| 32 | কুল | 70 | ছে | 108 | ইবেন | | |
| 33 | যূথ | 71 | ছি | 109 | ইল | | |
| 34 | মালা | 72 | ছেন | 110 | ইলে | | |
| 35 | রাজি | 73 | ছিস | 111 | ইলি | | |
| 36 | রাশি | 74 | ছিল | 112 | ইলেন | | |
| 37 | নিকর | 75 | ছিলি | 113 | ইলাম | | |
| 38 | নিচয় | 76 | ছিলে | 114 | ইলেম | | |

Figure 7. List of postfixes in Bangla

as it requires faster checking to verify whether any word is stopword or not. When a new stopword is required to be added, it is just appended in the corresponding table.

*5) Synonym Identifier:* Synonym identifier sub-module identifies the synonyms of words. In this paper, we used a special synonym handling mechanism to store the synonyms. It is explained below.

The creation of synonym module stores the synonym of words corresponds to each language. The synonyms are stored in the database scanning the words of document database excluding stop words. The document database is scanned twice to create the synonym list. After the first scan of the database initialization without the synonym effect, the $Occur$ table of the database as shown in Figure 5 contains different words of the document database. This table also contains words with their synonyms. So, this table is checked to find the group of words having same meaning.

With the group of words, a synonym table for each language is created as shown in Figure 6. In this structure, the words having same meaning will have same synonym id that forms a group of synonyms. In each group of synonyms, a sequence number is maintained according to the importance.

During the second scan of the database initialization the synonym table is used to keep the effect of synonyms. This module also manages the storage of the terms in such a way that the plagiarism detection system can run uniformly overcoming the synonym-handling problem. Storage module stores term information and necessary related information in the database according to Figure 5. In Figure 5, $keyword$ and $Doc\_ID$ of the $Word$ table represents which keyword presents in which document and $Freq$ field shows how many times the keyword occurs in that document. The $Doc\_ID$, $Title$ and $Size$ fields of the $Document$ table represent document ID of documents, Title and number of terms present in that document, respectively.

In $Occur$ table the field $Keyword$ and $Doc\_freq$ is used to represent in how many documents a keyword occurs. After stemming, the root is stored in the $Word$ table of the database with the corresponding document ID. All the keywords of all documents are stored in the $Word$ table. After the creation of $Word$ table, the $Occur$ table is created from $Word$ table. A record in $Document$ table is inserted after completion of scanning of each document.

*6) Keywords Extractor:* The keyword extractor sub-module extracts the root of every word using morphological analyzer. The algorithm for keyword processor takes words as input and gives the root of the words as output. Finding the root of a word is called stemming. We have used a list of 143 postfixes for Bangla as shown in Figure 7 collected from Bangla grammar books and 127 postfixes for English from [27]. Anyone can enrich the list. The overall procedure of keywords extraction in given in **Algorithm 2**.

To stem the postfixes from the terms of the document set, the morphological analyzer checks the terms against the postfix list of the corresponding language. Two important issues in keywords extraction include the selection of next possible alphabets of different postfixes to which next matching occurs and the identification of whether the current alphabet is the end of any postfix matching. It can be handled efficiently if we know the appropriate character sequence in the postfixes. As in our system all the terms are unicode supported, we consider the phonetic sequence of characters for their appearance in the terms. As for example, the sequence of characters in the postfixes of Figure 7 is given in Figure 8.

*7) Addition of New Documents:* For addition of a new document, first of all the maximum document ID is obtained from the $Document$ table. The new ID is incremented by one for the new document. Then the document's terms are scanned one by one until the end of the file is reached. For each term, it is checked whether it is in stoplist or not. If it is in the stoplist, it is just skipped. If it is not in the stoplist, morphological analysis is done to find the root of the word.

The root word is checked in the synonym table. If it is in the table, then the equivalent term is checked in the $Word$ table for that document whether it is present or not. If it is not present then one record is inserted into the $Word$ table with values equivalent synonym as keyword, the new document id as $Doc\_ID$ and frequency as 1. If it is present with the new document id then the Freq value of the term for that term will be incremented by 1. If the root word is not found in the synonym table, it is inserted into a temporary table with the same structure as the word table. In this way the scanning of the new document is

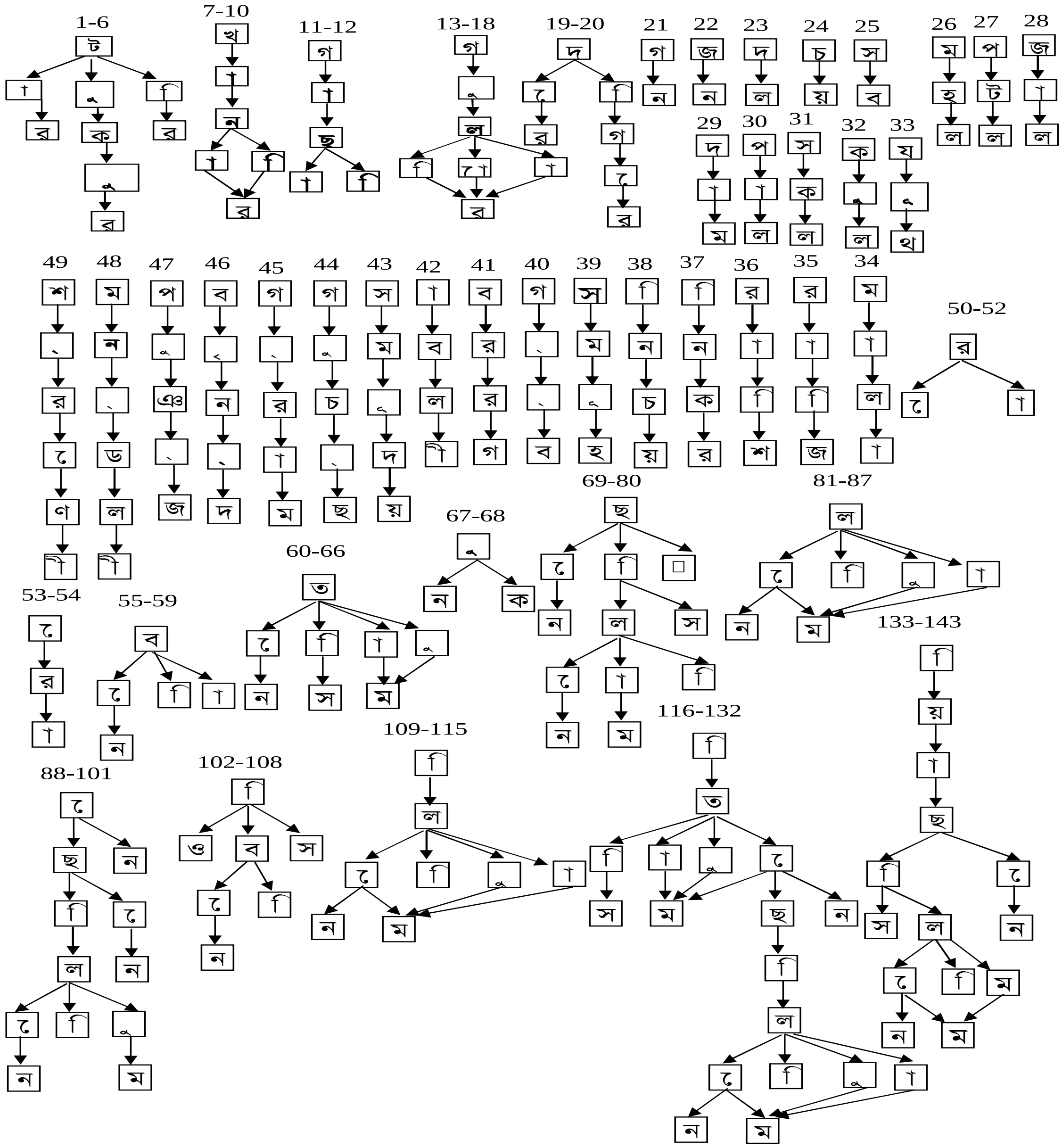


Figure 8. Character sequence of postfixes in Bangla

completed. If no term is found in the temporary table after the complete scan of the file the *Occur* table is processed from *Word* table according to **Algorithm 3**.

If there are some records in the temporary table then it is considered that the terms occurring in those records have no entity in the synonym table. As a result no effect of synonyms is considered for those terms. So synonym table is updated for the terms in temporary word table. **Algorithm 4** performs updating task of the synonym table.

**Algorithm 4** shows how the synonym is generated for the newly added document. Now it is required to add the terms from temporary word table to the *Word* table considering the effect of synonyms. Procedure to add the terms from temporary word table to word table is given **Algorithm 5**.

After inserting the term information into *Word* table, the *Occur* table is updated according to **Algorithm 1**. Then the *Document* table is updated for the new document with its size. In this way all the information regarding the new document is stored in the database.

*8) Bilingual Translator:* In the method of bilingual plagiarism detection, the translation of keywords between the documents is necessary. **Algorithm 6** performs the task of such translation. The algorithm first searches for the existence of the keyword in the dictionary. If the item exists in the dictionary the algorithm does not insert the item in the table. Otherwise, the algorithm inserts the item in the dictionary along with its translated values in other language. So, there is only one entry of a specific keyword in the dictionary.

**Algorithm 2** *KeywordsExtraction*

**Input:** List of terms $T = t_1, t_2, \cdots, t_x$ and list of postfixes $P = p_1, p_2, \cdots, p_y$ in a language
**Output:** Keywords

```
begin
for each term $t_i, 1 \le i \le x$ do
    Check each character of $t_i$ from last with the last character of $p_j, 1 \le j \le y$
    if a match is found then
        Check next character of $t_i$ with the character at same position of $p_j$ until a complete matching with $p_j$ is found
        Discard the matched part from $t_i$ and return remaining part as root $r_i$
    else
        Discard the scanning of the corresponding term and return $t_i$ as keyword $r_i$
    end if
end for
end
```

**Algorithm 3** *Update_occur_tbl_new_document*

```
begin
Select maximum document id from the word table
Get all the terms from the word table where document id = maximum document id
for each selected term do
    if the term is present in the occur table then
        Increment the doc_freq value of that term by 1 in the occur table
    else
        Insert the term into occur table with doc_freq value by 1.
    end if
end for
end
```

### B. Storage

Storage module stores information processed by database initialization and processing module. Bilingual dictionary keeps the mapping of keywords in two different languages.

### C. Query Execution Module

Query execution module takes the document that is necessary to check for plagiarism as input. The parse query sub-module calls synonym identifier, stoplist creator and keywords identifier sub-modules to generate the corresponding information from the query document. Then vector space model [28] is created and based on vector space model relevancy is calculated using cosine distance formula. Finally, retrieved relevant documents are sorted in descending order of their relevancies and return to the user via query response sub-module. Here, the user can restrict the system to return top $n$ relevant documents.

**Algorithm 4** *Update_synonym_tbl_new_document*

```
begin
Get all the terms from the temporary word table
Mark all those terms as unmarked.
for each unmarked term do
    Select maximum synonym id from the synonym table
    Increment the selected id by 1
    Insert the unmarked term into synonym table with id 1 and sequence number 1
    Scan all the other unmarked terms to find the synonyms of the previous term
    if some terms are found then
        for each such term do
            Insert the term into synonym table with the same id and sequence number as sequence number +1
            Mark all the terms as marked having the generated synonym id
        end for
    end if
end for
end
```

**Algorithm 5** *Insert_word_tbl_new_document*

```
begin
Select document id for the new document
Get all the terms from the temporary word table
for each selected term do
    Find the terms equivalent synonym from the synonym table having sequence number 1
    Check this equivalent synonym in word table with same doc_id
    if present then
        Increment the freq value of that term for the new document by 1
    else
        Insert a record in the word table with the equivalent synonym having the new document id and freq value as 1
    end if
end for
end
```

In query execution module when the relevant documents are listed they can be accessed from front end, because all the documents stored in the database are as BFILE. When a relevant document is needed to view, its corresponding Doc_ID is selected and the file corresponds to that Doc_ID is displayed in a graphical user interface.

### D. Complexity Analysis

The plagiarism detection system developed based on documents' overall contents has two parts for the time concern, one for storing the necessary information in index structure and another one is query time for any particular query. Here, we just give time complexity for

**Algorithm 6** *BilingualTranslation*

```
Input: Keywords in a language Require: Update of the dictionary if necessary
begin
for each keyword do
    search the dictionary
    if the keyword is found in the dictionary then
        do nothing
    else
        add the keyword in the database table with its translated value in other language
    end if
end for
end
```

relevancy check. We do not give the complexity analysis for storing the documents.

Let, $n$ be the number of keywords in the query documents, $D$ be the number of documents in collection, $Q$ be the time required to get necessary information about a keyword. So, time required for querying all the keywords in a query document is $O(nQ)$, time required for calculating relevance of $D$ number of documents is $O(D)$ , and time required for sorting the relevant documents is $O(DlogD)$. So, total time complexity for searching is $O(nQ) + O(DlogD)$.

## VI. BILINGUAL PLAGIARISM DETECTION BASED ON DOCUMENTS' STATISTICAL INFORMATION

Our statistical method for bilingual plagiarism detection mainly based on several stylometric features [29] of the documents. Stylometric features can measure different aspects of writing style, and can be useful for detecting plagiarism from the documents of the same domain. In our statistical approach, we have used four different stylometric features to generate statistical information of each document. These are (i) number of sentences, (ii) average sentence length, (iii) sentence type, and (iii) tense form use each sentence. We have used the concept of [30] for generating above stylometric features of each document. Here, the system does not consider the sentences with less than four words for analysis. This is because from the observation it has been found that most sentences with less than four words do not contain interesting information for comparison.

When a query document comes, the system generates similar statistical information of the query document.

Then based on the statistical information of the documents, following mathematical formula is used to calculate the relevancy between two documents.

$$\frac{S_{tr}-S_{te}}{S_{tr}}+\frac{L_{tr}-L_{te}}{L_{tr}}+\frac{Sim_{tr}-Sim_{te}}{Sim_{tr}}+ \\ +\frac{Cm_{tr}-Cm_{te}}{Cm_{tr}}+\frac{Cp_{tr}-Cp_{te}}{Cp_{tr}}+\frac{Pr_{tr}-Pr_{te}}{Pr_{tr}}+ \\ +\frac{Pa_{tr}-Pa_{te}}{Pa_{tr}}+\frac{Fu_{tr}-Fu_{te}}{Fu_{tr}} \quad (1)$$

In above equation, $te$ and $tr$ stand for test and training documents respectively. The meanings of remaining symbols are as $S$: total number of sentences, $L$: average sentence length, $Sim$: number of simple sentences, $Cm$: number of complex sentences, $Cp$: number of compound sentences, $Pr$: number of sentences with present tense, $Pa$: number of sentences with past tense and $Fu$: number of sentences with future tense.

From the above equation, it is observed that for any two documents calculated value closer to zero means high level of plagiarism and far from zero indicates low level of plagiarism. A value zero indicates full plagiarized document.

Similar to our previous method, in this approach, we can also restrict the system to return the documents below an user defined threshold and any relevant document can be viewed in a graphical user interface.

There are several limitations of this approach. Main limitation of this approach is that it just gives an approximation of plagiarization but not actual plagiarized information. Another limitation is that this approach is highly domain specific and the documents size need to be uniform. However, the main advantage of this method is that it is independent of the language in use. This method is well applicable at the university level to check the originality of the students' assignments.

## VII. EXPERIMENTAL RESULTS AND DISCUSSION

This section discusses the experimental setup for the simulation and the corresponding results.

### A. Experimental Setup

The plagiarism detection system has been developed on a machine having an Intel(R) Core2 Duo, 2 GHz CPU, and 3 GB main memory, running on Microsoft Windows XP operating system. The system was implemented in Jbuilder-8 in the front and Oracle 10g DBMS in the backend for storing the text database and related information. We have used total 110 documents for experimental purpose. These documents were collected from a department of a public university. Students were asked to submit their report individually from a specific domain. Total hundred and ten students were divided into two equal size groups. Students of one group submitted their reports in Bangla and another group submitted their reports in English. Among 110 documents, we have used 50 Bangla documents and 50 English documents as training documents. Remaining 10 documents were used for test purpose. The documents are almost equal in size.

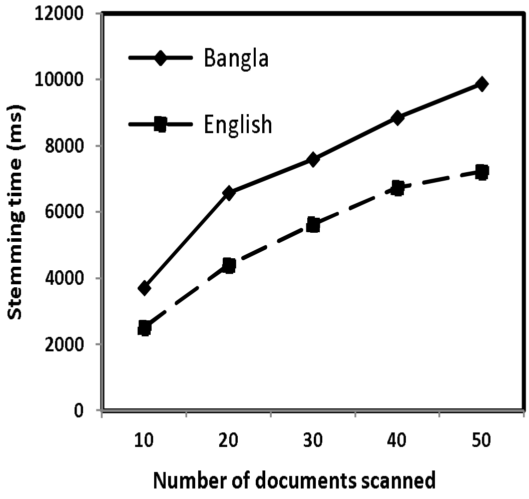


Figure 9. Stemming time varying number of documents

### B. Postfix Statistics and Stemming Performance

We analyzed the syntax of various words in Bangla documents using 143 postfixes and English documents against a list of 127 postfixes. In our experiment, we found that our data set requires stemming of around 62 different kinds of postfixes in Bangla documents and around 33 postfixes in English documents.

Then, we have analyzed the stemming performance. We performed stemming on the dataset initially with ten documents and incrementally added ten word documents in the system. We have performed stemming separately on English and Bangla documents data set. Figure 9 shows that the stemming time increases almost linearly with the increase of document number and stemming of Bangla documents require more time than English documents.

### C. Performance Based on Morphological Analysis

Next, we measured the performance based morphological analysis. We compared the result obtained by performing morphological analysis with the result obtained without morphological analysis. Figure 10, shows the result. We have used five Bangla and five English documents as test documents and the training corpus varies from twenty to hundred documents with almost equal number of Bangla and English documents. We set similarity percentage to thirty. We have performed query with five different documents of each language on same corpus and took the average number of retrieved documents from each training corpus. From Figure 10, it is observed that in both cases the relevancy is less when no morphological analysis is performed. This poor performance result is due to the words with postfixes resembles to different words and cannot contribute much in relevance calculation.

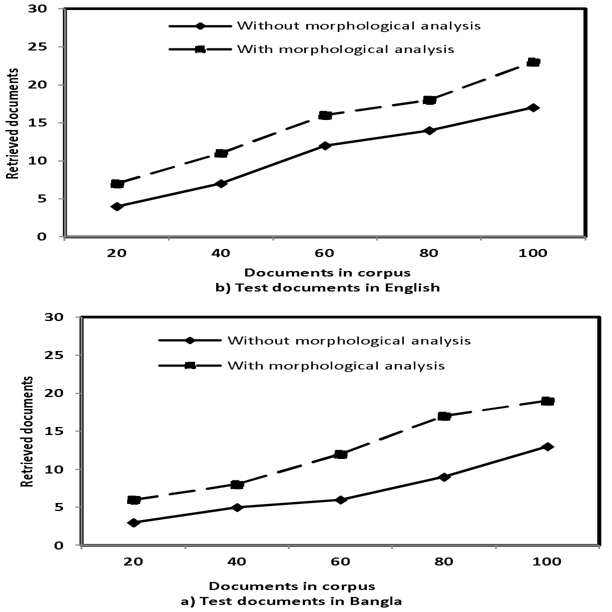


Figure 10. Comparative relevancy with morphological analysis and without morphological analysis

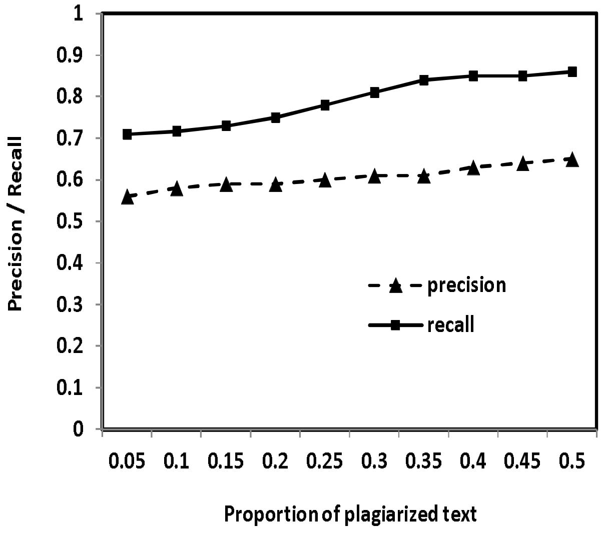


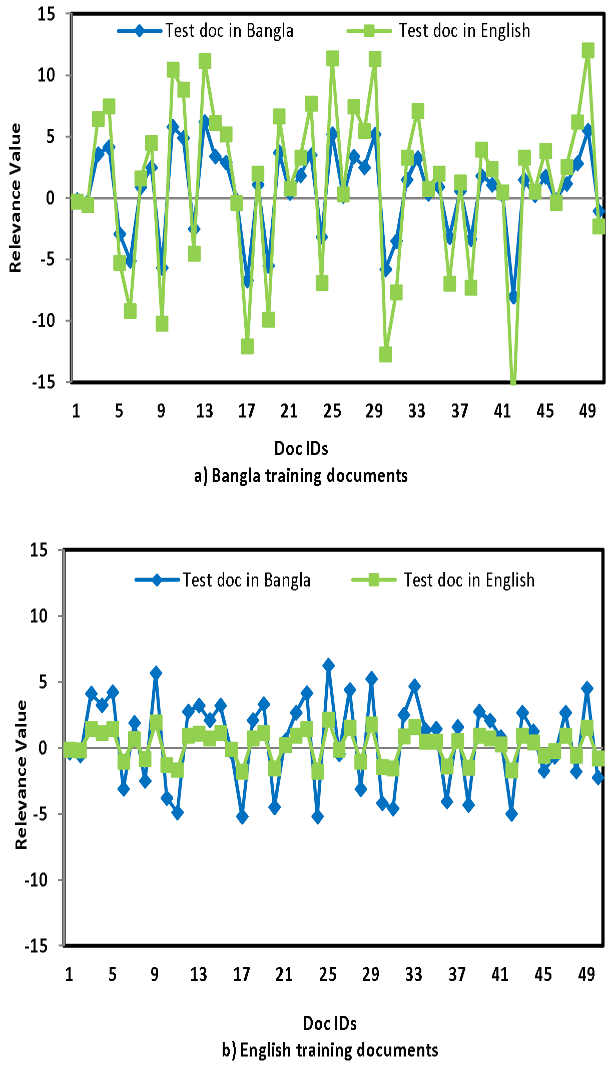
Figure 11. Performance analysis with increase level of plagiarism

### D. Performance Analysis Based on Severity of Plagiarism

In this experiment, we have modified our original training documents by replacing their contents with several plagiarized contents of different lengths. Then, we have analyzed the performance against the level of plagiarization. Figure 11 shows the result. From Figure 11, we can find that our system has good detection rate of plagiarism in terms of precision and recall with respect to the plagiarism severity.

### E. Performance Based on Documents Statistical Information

In this experiment, we have gathered the statistical information of each document. The information include number of sentences in each document, average length of sentences, number of sentences of each type, number of sentences based on tense of verbs. Here, we consider type of sentences as simple, complex and compound and



a) Bangla training documents

b) English training documents

Figure 12. Statistical analysis of relevancies between training and test documents

tenses as present, past tense and future tense. No sub-classification of tenses is considered. We use total ten test documents. The contents of five documents are in Bangla and the remaining five documents are in English. We perform plagiarization check with each of the ten test documents against each training document and then take the average of relevance values of five Bangla documents and five English documents separately. We used equation (1) for relevancy calculation.

Figure 12 shows an approximation of plagiarization in test documents from the training documents. Figure 12(a) shows the result when the training documents are in Bangla and Figure 12(b) shows the result when the contents of the training documents are in English. In first case, we use training corpus of 50 bangla documents and in second case, we use 50 English documents as training corpus.

In the result of Figure 12, a value closer to zero means high level of plagiarism and far from zero indicates low level of plagiarism. A value zero indicates full plagiarised document. From Figure 12, it is also observed that relevancy between training and test documents of same language are higher than the relevancy of documents in different languages. The change in the structure of sentences and other changes in the equivalent documents of two different languages is the main reason of such result.

## VIII. Conclusion

Detection of plagiarism from documents spanning over different languages is a major concern in current globalized world. While existing commercial and non-commercial plagiarism detectors provide good support for checking plagiarised documents within same language domain, they suffer from detecting plagiarism from documents of different language domains. In this paper, we have proposed two different approaches for detecting plagiarism from documents of two different languages. First method is based on removal of stop words, extraction of keywords, checking for synonyms and bilingual translation. The second approach is based on statistical information of the documents. We have implemented the systems for two languages: English and Bangla and found that the system can efficiently detect plagiarism from documents of both languages. Though we have considered Bangla and English languages for the experimental purpose, the system can adapt other languages with slight modification in the dictionary and storage of documents structure.

In this work, we consider an in-house database of documents and it is necessary to scan a document before it is considered in the system for plagiarism detection. In future, we hope to develop a system that can efficiently detect plagiarism from the documents in the Internet without scanning them in the system.

## References

[1] WriteCheck: Available: http://www.writecheck.com/
[2] EVE2: Available: http://www.canexus.com/eve/
[3] DOC Cop. Available: http://www.doccop.com/
[4] Plagium. Available: http://www.plagium.com/
[5] Plagiarism Detector. Available: http://www.plagiarism-detector.com/
[6] CodeMatch. Available: http://www.safe-corp.biz/products-codematch.htm
[7] Pl@giarism. Available: http://www.plagiarism.tk/
[8] Glatt Plagiarism Self-Detection Program (GPSD). Available: http://www.plagiarism.com/self.detect.htm
[9] Glatt Plagiarism Screening Program (GPSP). Available: http://www.plagiarism.com/screen.id.htm
[10] WCopyfind. Available: http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/
[11] Z. Ceska, M. Toman, and K. Jezek, Multilingual plagiarism detection , *In Proc. of 13th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, Varna, Bulgaria, September 2008, pp. 83-92.
[12] M. Potthast, A. B. Cedeno, B. Stein, and P. Rosso, Cross-Language plagiarism detection, *In the Journal of Language Resources and Evaluation*, vol. 45, no. 1, January 2010, pp. 45 - 62.
[13] A. B. Cedeno, P. Rosso, E. Agirre, and G. Labaka, Plagiarism detection across distant language pairs, *In Proc. of the 23rd International Conference on Computational Linguistics*, Beijing, china, August 2010, pp. 37-45.

[14] Plagiarism detection. Available: http://en.wikipedia.org/wiki/ Plagiarism_detection/

[15] T. C. Hoad and J. Zobe, Methods for identifying versioned and plagiarised documents, *Journal of the American Society for Information Science and Technology*, vol. 54, no. 3, pp. 203-215.

[16] B. Stein, Fuzzy-fingerprints for text-based information retrieval, *In Proc. of 5th International Conference on Knowledge Management*, 2005, pp. 572-579.

[17] K. Monostori, A. Zaslavsky, and H. Schmidt, Document overlap detection system for distributed digital libraries, *In Proc. of the Fifth ACM Conference on Digital Libraries*, 2000, pp. 226-227.

[18] B. S. Baker, On Finding duplication in strings and software, Technical Report, AT&T Bell Laboratories, Murray Hill, New Jersey.

[19] D. V. Khmelev and W. J. Teahan, A repetition based measure for verification of text collections and for text categorization, *In Proc. of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 104-110.

[20] A. Si, H. V. Leong, R. W. H. Law, CHECK: A document plagiarism detection system, *In Proc. of the ACM symposium on Applied computing*, 1997, pp. 70-77.

[21] H. Dreher, Automatic conceptual analysis for plagiarism detection, *In The Journal of Issues in Informing Science and Information Technology*, vol. 4, 2007, pp. 601-614.

[22] M. Zechner, M. Muhr, R. Kern, and M. Granitzer, External and intrinsic plagiarism detection using vector space models, *In Proc. of 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse and 1st International Competition on Plagiarism Detection*, 2009, pp. 47-55.

[23] B. Gipp, and J. Beel, Citation based plagiarism detection - a new approach to identify plagiarized work language independently, *In Proc. of the 21th ACM Conference on Hyptertext and Hypermedia*, June 2010, pp. 273-274.

[24] B. Gipp, N. Meuschke, and J. Beel, Comparative evaluation of text- and citation-based plagiarism detection approaches using GuttenPlag, *In Proc. of 11th ACM/IEEE-CS Joint Conference on Digital Libraries*, June 2011, Ottawa, Canada, pp. 255-258.

[25] B. Gipp, and N. Meuschke, Citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence, *In Proc. of the 11th ACM Symposium on Document Engineering*, 2011, pp. 249-258.

[26] D. I. Holmes, The evolution of stylometry in humanities scholarship, *In the Journal of Literary and Linguistic Computing*, vol. 13 no. 3, 1998, pp. 111-117.

[27] List of Suffixes. Available: http://www.examples-help.org.uk /definition-of-words/list-of-suffixes.htm/

[28] G. Salton, A. Wong, and C. S. Yang, A Vector space model for information retrieval, *In Communications of the ACM*, vol. 18, no. 11, November 1975, pp. 613-620.

[29] P. Juola, Authorship Attribution, *In Foundations and Trends in Information Retrieval*, vol. 1, no. 3, 2006, pp. 233-334.

[30] S. Azad, and M. S. Arefin, Bangla documents analyzer, *In Proc. of International Conference on Electronics, Computer and Communication*, Rajshahi, Bangladesh, June 2008, pp. 438-442.

**Mohammad Shamsul Arefin** received his B.Sc. Engineering in Computer Science and Engineering from Khulna University, Khulna, Bangladesh in 2002, and completed his M.Sc. Engineering in Computer Science and Engineering in 2008 from Bangladesh University of Engineering and Technology (BUET), Bangladesh. Now he is a PhD candidate at Hiroshima University with support of the scholarship of MEXT, Japan.

He is a member of Institution of Engineers Bangladesh (IEB) and currently working as an Assistant Professor in the Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh. His research interest includes privacy preserving data mining, multilingual data management, semantic web, and object oriented system development.

**Yasuhiko Morimoto** is an Associate Professor at Hiroshima University. He received B.E., M.E., and Ph.D. from Hiroshima University in 1989, 1991, and 2002, respectively. From 1991 to 2002, he had been with IBM Tokyo Research Laboratory where he worked for data mining project and multimedia database project. Since 2002, he has been with Hiroshima University. His current research interests include data mining, machine learning, geographic information system, and privacy preserving information retrieval.

**Mohammad Amir Sharif** received his B.Sc. Engineering in Computer Science and Engineering from Khulna University, Khulna, Bangladesh in 2002, and completed his M.Sc. Engineering in Computer Science and Engineering in 2007 from Bangladesh University of Engineering and Technology (BUET), Bangladesh. Now he is a PhD candidate University of Louisiana at Lafayette, USA.

Currently he is working as an Assistant Professor in Mawlana Bhashani Science and Technology University, Tangail, Bangladesh, in Information and Communication Technology Department. His research interest includes information retrieval, distributed computing, multilingual data management, and Intelligent systems.