**Project Proposal**

**External Plagiarism Detection**



Submitted to the

Project Management Committee

Department of Computer Science & Information Technology

University Of Sargodha, Mandi Bahauddin Campus

Submitted By

|  |  |  |
| --- | --- | --- |
| Sadia Parveen | BSIT-F14-34 | |
|  | |  | |

University of Sargodha, Mandi Bahauddin Campus

Faculty of Computer Science & Information Technology

Faculty of Computer Science & Information Technology

APPROVALS

Project ID (for office use) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Session: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Agree to Adviser: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Name of adviser)

Date Submitted: 13-11-2017

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date Approved: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1**. **Introduction……….…………………………………………….……………………1**

1. 1 Project Title…….……………………………………………………………1

1.2 Introduction and Background…...……………………………………………2

1.3 Problem Statement……...……………………………………………………2

1.4 Previous Work…..……………………………………………………………3

1.5Objectives of the Project ……………………………………………………3

1.6 Specific Project Goals…..……………………………………………………3

1.7 Scope of the Project (Abstract)………………………………………………4

1.8 Glossary ……………………………………………………………………4

1.9 Project Overview Statement Template….……………………………………4

**2. Benefits of the Project……………………………………………………………..…5**

2.1 Direct Customers / Beneficiaries of the Project ……………………………5

2.2 Outputs Expected from the Project……...……………………………………5

**3. Project Description ……..……………………………………………………………5**

3.1 Functional Specification ……………………………………………………5

3.1.1 Functions Performed ……………………………………………6

3.1.2 Limitations and Restrictions…..……………………………………6

3.1.3 Application Architecture……...……………………………………7

3.1.4 Equipment Configuration……..……………………………………7

3.1.5 Implementation Tools and Technology ……………………………7

3.2 Implementation Plan…………………………………………………………8

3.2.1 Deliverable Items.......................................................................…...8

3.3 Project Schedule / Milestone Chart……..……………………………………8

**4. Reference ……………………………………………………………………………8**

## 1 Introduction

In recent years plagiarism and it’s detection got significant attentions in both industry and academia .There are many research papers written on plagiarism using different tool and techniques for efficient and accurate detection. Each tool and techniques have their own advantages and limitations. But the task plagiarism detection itself divided into two main categories:

(1) External plagiarism detection and (2) intrinsic plagiarism detection. The goal of ***external plagiarism detection***is to identify the Source (or original) document(s) that has been used to plagiarism a suspicious document. On the other hand, in ***intrinsic plagiarism detection*** the source documents are not available and plagiarized text is identify by looking for stylistic inconsistencies or text which is different from the rest.

Our concern area is external plagiarism detection.

## 1.1 Project Title

***External plagiarism detection using natural language processing technique*** is the title of this project. Plagiarism comes in many forms. It can happen in any field that involves creation process, which includes written text, computer source code, art and design, and even music pieces.

The Types of plagiarism which has been addressed in previous research are mainly:

1. Multiple-choice tests.
2. Source code in programming languages.
3. Written text, also known as free text and natural language text plagiarism.

Plagiarism in multiple-choice test and source code are different from natural language text plagiarism. Detecting plagiarism in multiple-choice relies on some statistical methods. On the other hand the source code detection require different tools and metrics which captures statistical features to determine similarities between the codes.

In this project, the focus is on written text as it poses a greater challenge. It involves both linguistic features along with statistical features. For written text plagiarism, the most common cases are found in academic settings. The following types are examples of how plagiarism can occur in academia:

* Submitting someone else’s work
* Insufficient referencing
* Direct copying, from one or multiple sources
* Paraphrasing

The above cases can occur in two types of text.

1. **Monolingual** (copied from one language)
2. **Cross Lingual**(copied from a second language, sometimes known as translated  
   plagiarism)

In ***monolingual*** ***detection*** the suspicious case and source case belong to the same language.

The ***Cross Lingual detection*** is needed when the suspicious case are derived from the source case of different language.

So our major concern is on external detection on natural language text plagiarism using monolingual techniques.

## 1.2 Introduction and Background

In the early days, plagiarism was only be detected manually by relying on the readers’ own knowledge. As the cognition factor varies between person to person, and the vast amount of material is impossible to attain, the process of identifying plagiarism within text can be difficult task. The obvious disadvantage of manual method is that when the amount of information increase, a reader is less likely to be able to identify the similarities.

With the passage of time different tools, technologies and method are used to perform this task. These are a lot efficient from manual method. Every method and tools have their own benefits and drawbacks discussed in previous work.

**1.3** **Problem Statement**

There are many factors which mostly unsolved. The problem statements are based on the characteristics of plagiarism which are not met yet.

The difference between the proposed system and traditional system are as follows.

1. Lexical changes
2. Syntactic changes
3. Semantic changes
4. ***Lexical changes*** involve addition deletion or replacement of words in the text. For example a sudden changes of vocabulary, such as excessive use of new terminology within a document which is usually good indication of copy and paste plagiarism. Another example is word for word substitutions by synonyms. This type of plagiarism is undetectable using the traditional string-matching approach. Detection would require the analysis of *lexical information* throughout the text.
5. ***Syntactic changes*** involve rearrangement of the structure of the text. Their examples are active voice and passive voice and word/clause reordering etc. Similarities in syntactic structures can be an indication of plagiarism, but again it is undetectable using the traditional string-matching approach, and detection would require the analysis of syntactical structure of text.
6. ***Semantic change*** involves more radical changes in the text, normally based on heavy paraphrasing that can include both lexical and syntactic changes. Detecting this type of change would require the analysis of semantic information to judge whether two texts hold the same meaning. Again, this is undetectable with the traditional approaches.

**1.4 Previous Work**

Plagiarism detection systems started off as detection tools for multiple-choice tests and computer source code. Plagiarism detection systems for natural language were not developed until the 1990s.

Between 1990 and 2000, most systems developed were intended for detecting programming code plagiarism, and only a handful of researches focused on  
plagiarism detection for written texts. An example of these early written text  
detection approaches was a prototype, COPS. It was designed to detect complete  
or partial copies of digital documents. The similarity between documents was measured by using sentence-level matching. The sequences of sentences in each document were matched against other sequences in documents in the dataset. However, the sentence-based approach was not effective in detecting partial sentence overlaps.

As an extension to **COPS**, Shivakumar and Garcia-Molina (1995, 1996) proposed a prototype **SCAM**. The approach introduced the removal of stop words and  
frequent words as a pre-processing step. The texts were compared as overlapping  
sequences of words or sentences, and thresholds were set to determine three levels  
of overlap between texts: exact copies, high overlap and some overlap.

Another example of early research, the YAP3 tool for identifying similarities in programming code, utilized the Running-Karp-Rabin Greedy-String- Tiling (RKR-GST) algorithm as a structured-metric similarity detection system. The **RKR-GST** algorithm is a variant of the Longest Common Subsequence (**LCS**) algorithm which allows a maximal match alongside a minimal match length between texts. This algorithm was introduced to handle cases where sequences of texts had been reordered.

By the end of the decade in **2000**, only a handful of commercial approaches were available for written text plagiarism detection, for example **EVE2** and **I Paradigms** (the early version of Turnitin.org). Both approaches perform online detection by searching for similar texts on the Internet, and offline detection by comparing texts with their own database.

Between 2000 and 2012, the field saw a huge surge of new plagiarism detection methodologies and their implementations. From 2000 onwards, more and more research began to address the issue of written text plagiarism detection.

Kang et al. (2006) developed another *structural metric*, Plagiarism Pattern Checker, which checks for plagiarism patterns by measuring the amount of over lapping n-grams within a sentence. It also incorporates Word Net to check for synonyms.

Fingerprinting methods started to gain attention as the amount of information available increased. This method is said to be much more efficient as it generates a hashed description, which is the “fingerprint” for each document, and then the fingerprints of documents can be compared instead of the entire document.

Other advanced approaches started to emerge by 2007. The research suggested using a semantic-based approach for plagiarism detection, by combining an information retrieval model based on tf-idf with latent semantic indexing (LSI).

In 2009 text pre-processing techniques, such as stop word removal, and shallow NLP techniques, such as stemming, are applied to documents before computing similarity. Short sentences are also removed.

**1.5 Objectives of the Project**

The objectives of the project are minimized the limitations of the existing systems. Like lexical changes, syntactic changes and semantic changes are not detected by the existing systems.

* To recognized the reordered structure.
* To recognized the synonyms.

**1.6** **Specific Project Goals**

The main aim of this project is to investigate the use of Natural Language Processing (NLP) techniques in text reuse detection and direction identification tasks.  
The hypothesis is that original texts and rewritten texts exhibit significant and  
measurable differences, and that these can be captured through statistical and  
linguistic indicators. To test the hypothesis, a framework which incorporates NLP  
techniques along with existing shallow techniques is proposed to improve the identification of plagiarized texts.

## 1.7 Scope the Project (Abstract):

The scope of the project is limited to external plagiarism, where both the suspicious plagiarized texts and the potential source texts are available. All plagiarized texts and source texts are monolingual English written texts along with able to execute some sort of cross lingual text. The evaluation is based on an empirical corpus-based approach, where different corpora are used to test various experimental settings. When applicable, supervised machine learning models are used for text classification. Initial experiments refer to plagiarized text cases as “plagiarized documents”, which in later experiments are referred to as “plagiarized passages” as the plagiarized text length changes.

**1.8 Glossary**

NLP (natural language processing), LCS (Longest Common Subsequence), LSI (latent semantic indexing), RKR-GST (Running-Karp-Rabin Greedy-String-Tiling), External plagiarism detection, String-matching method, Monolingual, Cross lingual.

1.9 Project Overview Statement Template

|  |
| --- |
| Project Title:  External plagiarism detection |
| Group Leader: Sadia Parveen |
| Project Members:   |  |  |  |  | | --- | --- | --- | --- | | Name | Registration # | Email Address | Signature | | Sadia Parveen | BSIT-F14-34 | Sadiagondal123@gmail.com |  | |
| **Project Goal:**  The goal of my project is to create a tool which detect the plagiarism successfully and perform working properly using NLP technique.  This tool performs work offline as well as online. |
| **Objectives:**   |  |  | | --- | --- | | Sr.# | Objectives | | 1 | Tool able to detect synonyms. | | 2 | Tool can be able to detect syntactic changes. | | 3 | The result should be visualize in the form of graphs. | | 4 | Tool able to detect monolingual plagiarism as well as cross lingual. | | 5 | The graphs represent the plagiarism in percentage. | |
| Risks of the Project: ***(Please describe the factors that may cause delays in, or prevent implementation of, the project as proposed above; estimate the degree of risk.)***   |  |  |  |  | | --- | --- | --- | --- | | (Please mark 🗹 where applicable) | Low | Medium | High | | * Technical risk |  |  | 🗹 | | * Timing risk |  | 🗹 |  | | * Budget risk |  | 🗹 |  | |
| Organization Address (if any): |
| Type of project: 📺Research 🗹Development |
| Target End users:  Academic(Teacher, Students) ,industrial (researchers, translators) |
| Development Technology: 🗹Object Oriented 📺Structured |
| Platform: 📺Web based 📺Distributed  📺Desktop based 📺Setup Configurations  🗹Can be able to work online and offline\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Suggested Project Supervisor: Prof. Qaiser Farooq |
| Approved By: |
| Date:13-11-2017 |

## 2. Benefits of the Project

**2.1 Direct Customers / Beneficiaries of the Project**

Teachers, Students, Researchers.

* 1. **Outputs Expected from the Project**

A successful tool for detecting external plagiarism.

## 3. Project Description

The portion describe the functionality of the system and the services provided by the system to their customers.

**3.1 Functional Specification**

The user can perform comparison between two documents in offline scenario.

The user input both document to check the plagiarism between them. After the processing the user gets respond against their input.

In online scenario the user can input their document in two ways:

* User can place their text in the text field to check the plagiarism.
* User can attach their document on which the checking will perform.

After that the press the analyze button then the system respond them and evaluate the result against the document or text.

* + 1. **Functions Performed**

The framework for external plagiarism detection involves five stages.

***Pre-processing***:

This stage prepares the input text collection, including both suspicious and source texts, for subsequent stages. Text per-processing and shallow NLP techniques are applied to the texts.

***Similarity comparison:***

This stage performs pair-wise comparisons between each suspicious text against all source texts. One or more similarity metrics are applied to give each suspicious-source text pair a similarity score.

***Filtering:***

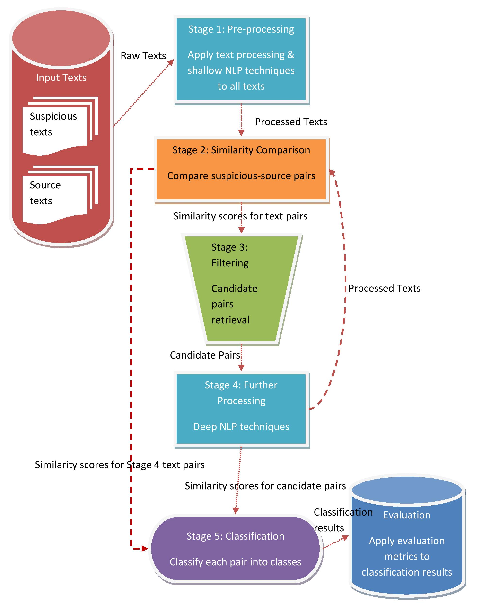
The similarity scores generated in stage 2 are used to judge the likelihood of a suspicious-source pair being listed as a candidate pair. The likelihood is determined by setting a threshold on the similarity scores. This can be done either by using a machine learning algorithm to learn the threshold, or by manually defining such a threshold. If a pair has reached a certain threshold, the pair is listed as a candidate pair; otherwise the pair is discarded as not plagiarized.

***Further processing:***

As deep linguistic features are computationally expensive, this stage is only applied to candidate pairs. Candidate pairs from Stage 3 are further processed; then Stage 2 is repeated for the pairs of Stage 4 to generate a similarity score.

**Classification:**

The final stage is to use the similarity scores from the  
previous stage to assign each text pair a classification as Plagiarized or Clean.  
In some cases the class Plagiarized can be further defined at various levels,   
such as Near Copy, Heavy Revision, or Light Revision.



Figure

**3.1.2 Limitations and Restrictions**

1. The scope of the system is limited to external detection.
2. The system does not support the cross lingual plagiarism.
3. High Technical Risk involve

## 3.1.3 Application Architecture

External plagiarism detection follow MVC pattern. MVC divide the system in three parts.

* **Model:**

This part represents the database. The data handling the tool.

* **View:**

This part represents the interface of the system.

* **Controller:**

This part made interaction between view and model.

**3.1.4 Equipment Configuration**

Windows operating system is required because the tool depends on this operating system.

Memory: 4GB

Internet Connection.

Hard Disk: 80GB

Processor: At Least Dual Core

**3.1.5 Implementation Tools and Technology**

|  |  |  |
| --- | --- | --- |
| Tools | Description | Usage |
| **NLTK** | Python NLP package | POS tagging, chunking, lemmatization, stemming |
| **Stanford** **CoreNLP** | Java NLP analysis tools | POS tagging, NER, dependency parsing, lemmatization, coreference resolution |
| **WordNet** | Lexical database | Lexical generalization |
| **VerbNet** | Class-based verb lexicon | Predicate generalization |
| **METEOR** | Machine translation evaluation system | Segment alignment (based on exact  word, stem, synonym and paraphrase) |
| **Weka** | Java software for machine learning | Machine learning, data mining |
| **VENSES** | SWI-prolog semantic evaluation system for recognizing textual entailment | Functional and syntactic constituency, topic identification (main, secondary, potential topics ) |
| **SENNA** | ANSI C NLP predictions | POS tagging, chunking, NER, semantic role labeling, syntactic parsing |

***3.2 Implementation Plan***

**3.2.1 Deliverable Items**

The Tool that fulfill the needs of customers from both perspective:

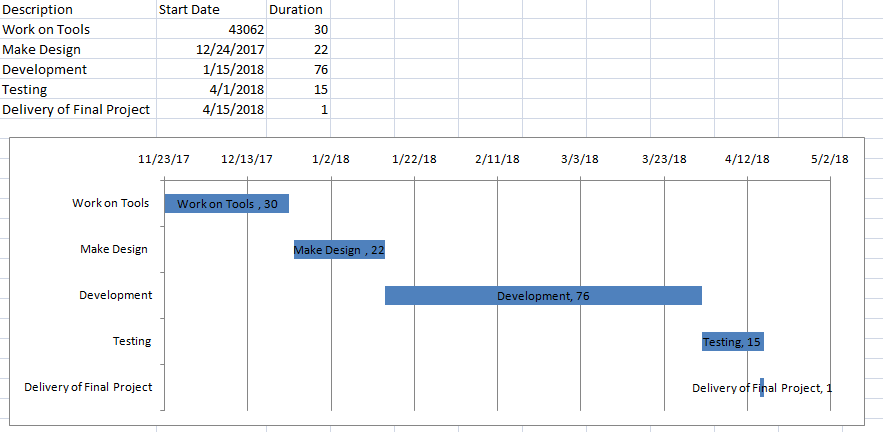
User Perspective:

* Input their data.
* Analyze them.
* Got responsive output.
* Can Support a large amount of quantity.

System Perspective:

* Collect the data(tokenization, lowercasing)
* Filtering(select the threshold)
* Detection(perform classification)

**3.3 Project Schedule / Milestone Chart**

****

***4. References:***

* <https://www.slideshare.net/NimishaT1/plagiarism-detection-tools-and-techniques>
* <http://ieeexplore.ieee.org/document/6993398/>
* <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=C5390DC059B46BAF68E0EAA77AC3A3D4?doi=10.1.1.474.5755&rep=rep1&type=pdf>