



**TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING  
THAPATHALI CAMPUS**

**Minor Project Midterm Report on  
On  
Plagiarism Detection in Nepali Theses**

**Submitted By:**

Ankit B.K. (THA076BCT006)  
Ayush Batala (THA076BCT011)  
Mishan Thapa Kshetri (THA076BCT019)  
Nishant Uprety (THA076BCT023)

**Submitted To:**

Department of Electronics and Computer Engineering  
Thapathali Campus  
Kathmandu, Nepal

February, 2023



**TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING  
THAPATHALI CAMPUS**

**Minor Project Midterm Report on  
On  
Plagiarism Detection in Nepali Theses**

**Submitted By:**

Ankit B.K. (THA076BCT006)  
Ayush Batala (THA076BCT011)  
Mishan Thapa Kshetri (THA076BCT019)  
Nishant Uprety (THA076BCT023)

**Submitted To:**

Department of Electronics and Computer Engineering  
Thapathali Campus  
Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor's Degree in  
Computer Engineering.

**Under the Supervision of**

Er. Saroj Shakya

February, 2023

## **ACKNOWLEDGEMENT**

We would like to express our sincere gratitude towards the Institute of Engineering, Tribhuvan University for the inclusion of minor project in the course of Bachelors in Computer Engineering. We are also thankful towards our Department of Electronics and Computer Engineering for the proper orientation and guidance during the project **“Plagiarism Detection in Nepali Theses”**

We would like to acknowledge the authors of various research papers and developers of various programming libraries and frameworks that we have referenced for developing our project. We would like to express our gratitude towards our Project Supervisor Er. Saroj Shakya for continuous suggestions throughout.

Finally, we would like to thank all the people who are directly or indirectly related during our study and preparation of this project.

Ankit B.K. (THA076BCT006)

Ayush Batala (THA076BCT011)

Mishan Thapa Kshetri (THA076BCT019)

Nishant Uprety (THA076BCT023)

## **ABSTRACT**

Plagiarism is the way of copying someone's works entirely or modifying it by using similar word and claiming it as their own. It has been a major problem in academics and literature field from the past. This practice in detecting plagiarism in Nepali Language hasn't been common as most of Nepali literature cannot be found in digitalized forms. Through this project we are going to develop a training model to detect the plagiarism in suspicious theses using natural language processing (NLP). We will be preprocessing the theses thoroughly and parse it into paragraphs then into sentences and tokens and then use different mathematical formulae like extended Jaccard similarity to assign the certain value to them. These values are used as inputs in training model using SVM to detect plagiarism. We are expecting to get the plagiarism percentage in document taking in account the citations

*Keywords: Extended Jaccard Similarity, NLP, Plagiarism, Support Vector Machine*

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT.....</b>	<b>i</b>
<b>ABSTRACT .....</b>	<b>ii</b>
<b>List of Figures.....</b>	<b>vi</b>
<b>List of Tables .....</b>	<b>vii</b>
<b>List of Abbreviations .....</b>	<b>viii</b>
<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 Background.....	1
1.1.1 Natural Language Processing .....	1
1.1.2 Support Vector Machine .....	1
1.2 Motivation .....	2
1.3 Problem Definition .....	2
1.4 Objectives .....	2
1.5 Project Application and Scope .....	2
<b>2. LITERATURE REVIEW .....</b>	<b>4</b>
<b>3. DATSET PREPARATION AND ANALYSIS.....</b>	<b>6</b>
3.1 Collection of Nepali theses from TUCL.....	6
3.2 Dataset Preparation.....	7
3.2.1 Standard Rules Regarding Citation.....	7
3.3 Dataset Analysis .....	8
3.3.1 Complete Plagiarism.....	8
3.3.2 Direct Plagiarism .....	8
3.3.3 Paraphrasing Plagiarism.....	9
3.3.4 Non-Plagiarized texts.....	10
<b>4. METHODOLOGY .....</b>	<b>11</b>
4.1 NLP Approach.....	11
4.1.1 Corpora of Plagiarized Texts .....	11

4.2	Procedural Sequence.....	11
4.2.1	Data Collection and Preprocessing .....	11
4.2.2	Model Building and Machine Training.....	12
4.3	Use Case Diagram .....	14
4.4	System Architecture .....	15
4.4.1	Feature Extraction.....	16
4.5	Flowchart of Proposed System .....	18
4.6	Instrumentation Tools.....	19
4.6.1	Programming Language.....	19
<b>5.</b>	<b>IMPLEMENTATION DETAILS .....</b>	<b>20</b>
5.1	Dataset Preparation.....	20
5.2	Text Preprocessing .....	21
5.3	Feature Extraction.....	23
5.4	Model Training and Testing .....	24
5.5	Model Deployment.....	24
<b>6.</b>	<b>RESULT AND ANALYSIS .....</b>	<b>26</b>
6.1	Accuracy with features .....	26
<b>7.</b>	<b>REQUIREMENT ANALYSIS .....</b>	<b>27</b>
7.1	Feasibility Analysis .....	27
7.1.1	Economic feasibility .....	27
7.1.2	Technical feasibility.....	27
7.1.3	Operational feasibility.....	27
<b>8.</b>	<b>REMAINING TASKS.....</b>	<b>28</b>
8.1	Completion of Dataset Preparation.....	28
8.2	Lemmatization of Nepali Words .....	28
8.3	POS tagging in Nepali Language .....	29
8.4	Computation of Word Similarity using Nepali Dictionary .....	29

<b>9. APPENDICES .....</b>	<b>30</b>
Appendix A: Project Timeline .....	30
Appendix B: Nepali Dataset Snippets .....	30
Appendix C: Code Snippets .....	31
<b>References .....</b>	<b>33</b>

## List of Figures

Figure 3-1: Collection of Nepali theses .....	6
Figure 4-1: Hyperplane .....	13
Figure 4-2: SVM model .....	14
Figure 4-3: Use Case Diagram .....	15
Figure 4-4: System Architecture .....	16
Figure 4-5: Flowchart of Proposed System .....	18
Figure 5-1: Dataset Distribution Chart .....	21
Figure 5-2: Application Interface .....	25
Figure 5-3: Result page .....	25
Figure 6-1: Accuracy plot vs Features .....	26



## List of Tables

Table 3-1: Number of Datasets .....	10
Table 9-1: Gantt chart .....	30

## List of Abbreviations

IOE	Institute of Engineering
LSA	Latent Semantic Analysis
ML	Machine Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Parts of Speech
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TU	Tribhuvan University
TUCL	Tribhuvan University Central Library

# **1. INTRODUCTION**

## **1.1 Background**

The origin of the word plagiarism is supposed to be from a Latin word “plagiarius” (literally meaning kidnapping) to denote someone stealing someone else’s creative work [1]. Plagiarism is typically not itself a crime but like counterfeiting. In academia and industry, it’s a serious ethical offense. Plagiarism is presenting someone else’s work or ideas as your own, with or without their consent, by incorporating it into your work without full acknowledgement. Plagiarism occurs in varieties of context including books, music, software, academic papers, journals etc. The ease of information sharing using internet has encouraged searching for literature as well as other ideas online to replicate other’s work as their own.

With more people using internet maintaining academic integrity in school and institution is becoming increasingly difficult. However, the success of these systems for detecting plagiarism based on their capacity to identify various frauds for modifying the texts without altering it’s meaning has been questionable ever since.

### **1.1.1 Natural Language Processing**

Natural Language Processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding. [2] NLP involves ability to understand text and spoken words by the computers. Only few NLP works in Nepali language has been performed mostly due to the increased complexity and lack of available resources which isn’t sufficient.

### **1.1.2 Support Vector Machine**

Support Vector Machine (SVM), is one of best machine learning algorithm used mostly for pattern recognition. Pattern recognition aims to classify data based on either a prior knowledge or statistical information extracted from raw data, which is a powerful tool in data separation in many classes. SVM is a supervised type of machine learning algorithm in which, given a set of training examples, each labelled as belonging to one

of the many categories, an SVM training algorithm builds a model that predicts the category of the new example. SVM has the greater ability to generalize the problem, which is the goal in statistical learning. We use this algorithm to classify texts into two classes either plagiarized or not. [3]

## **1.2 Motivation**

In universities and colleges, many students refer to many books and internet for their work, sometimes they just find easy to use someone's work/idea without properly crediting them especially in Nepali language where this practice hasn't been observed. This threatens the integrity of someone's honest work and provide false credentials to an individual for actions they did not perform.

Plagiarism Detection techniques in English language are common but very few in Nepali language mostly due to the increased complexity and lack of available resources. So, we have decided to work on a Plagiarism detection tool for Nepali language

## **1.3 Problem Definition**

The problem is the lack of plagiarism detection in Nepali language. This project aims to detect plagiarism effectively in Nepali theses. We have to prepare dataset in Nepali language for this project from the available raw theses from TUCL. We will be following principle of citing sources to check for plagiarism in theses.

## **1.4 Objectives**

The main objectives of our minor project are listed below

- To prepare dataset for Plagiarism Detection in Nepali language.
- To build a functioning prototype of plagiarism detection software.

## **1.5 Project Application and Scope**

Plagiarism is one of the major issues in academics and journalism. In academics, students and professor faces suspension and may losses their license if their documents are found to be plagiarized. While in journalism, reporters work needs to be trustful and

if they are found to copy the work of others, they may face legal charges. So, plagiarism detection algorithm can be used in those cases to verify their works.

The applications and scope of plagiarism detection are:

- Verification of research work done by the students and professors in academics.
- Approval of honest work from reporter in journalism.
- Fails to identify common knowledge and generally accepted or observable facts
- Doesn't consider plagiarism in diagrams, illustrations, charts, pictures, or other visual materials.

## 2. LITERATURE REVIEW

Plagiarism constitutes a threat to the educational process because students may receive credit for someone else's work or complete courses without actually achieving the desired learning outcomes. There are lots of plagiarism checker tools (e.g., Turnitin, Eve2, CopyCatGold, etc.), still plagiarism detection is a difficult task because of the huge amount of information available online [4]. However, none of the plagiarism detection tools are used to check plagiarism for Nepali Language documents.

Ram Bhakta Bahachan and Arun Kumar Timilsina proposed a Nepali Plagiarism Detection Framework using Monte Carlo Based Artificial Neural Network (MCANN). In this paper, authors collected different Nepali documents from different sources and passed it in the framework. The framework applied Cosine Similarity and Jaccard Similarity between each paragraph vector from the source and suspicious data. The obtained results were further analyzed for their accuracy. The mean accuracy for MCANN was found to be in the range of 98.65% and 99.86% during paragraph based and line-based comparison of the documents. [5]

Bal Krishna Bal proposed a rule-based recursive stemming algorithm for Nepali Plagiarism Detection. In this paper, authors created a stemming and lemmatization algorithm to pre-process Devanagari scripts more effectively and ultimately use it for detecting plagiarism between Nepali datasets. They also made the use of Jaccard Similarity in order to calculate similarity between the token of the preprocessed documents. The obtained accuracy and precision of the document at lexical level was found to be 95% and 93.33% [6]

Research in lemmatization had been conducted which was able to obtain the total accuracy of 70.10%. This algorithm was tested for Nepali Language which is based on Devanagari Script. The approach has given better result in comparison to traditional rule based system particularly for Nepali Language only. The main reason for lower accuracy was due to lack of availability of corpus of Nepali data during the research [7]

A report on Nepali Grammar concluded a brief overview of the sentential structure of Nepali Language with illustration. The parts of speech of Nepali language were discussed followed by a detailed discussion on the phrase structure of Nepali Grammar on the paper which could be the basis for POS tagging in Nepali language [8]

Support Vector Machine is a supervised machine learning algorithm which analyzes data and recognizes pattern classification problems. SVM has the greater ability to generalize the problem and predict the effective solution of new problems. The main idea of SVM is to construct an optimal hyperplane which can be used for classification. The performance given by the SVM is comparatively higher. It scales relatively well to highly dimensional data and the trade-off between classifier complexity and the trade-off and error can be controlled explicitly. The weakness includes the need of a good kernel-function. The major strength of SVM is that the training of the data is relatively easy [3]

Christopher Burges proposed extrinsic plagiarism detection using SVM in English Language. In this paper, they have used features like word pairs, word similarity, fingerprint similarity, LSA similarity. For classification, the data is preprocessed and feature extraction of each existing data on the training data is performed. Finally, the classification of the inputs is performed. It uses the input data as instances that have features' values that are calculated in the feature extraction process. Learning algorithm performs classification using the learning model generated by the modeling subsystem. The result of the classification is the decision whether the input data is plagiarized or not. The accuracy of the model created using SVM was found to be 84.375%. [9]

Zdenek Ceska proposed a new plagiarism tool called SVDPlag which employs Singular Value Decomposition. To examine the efficiency, the experiment used corpus of 950 text documents and indicated that this approach significantly improved the accuracy of plagiarism detection [10]

### 3. DATSET PREPARATION AND ANALYSIS

In this project we mainly focus on detection of external plagiarism i.e., when both source text and suspicious text are present using Support Vector Machine which is a supervised learning algorithm. We will be needed thousands of annotated Nepali texts to train the model. Due to complexity of task and lack of digitalized resources, plagiarism detection has not been performed in Nepali language [11]. The primary objective of the minor project has been to assemble a precise and reputable dataset for the purpose of training the model.

#### 3.1 Collection of Nepali theses from TUCL

Due to lack of practice of plagiarism detection in Nepali literature we were unable to find annotated dataset. We were able to find corpora of Nepali language thesis by student of Department of Nepali Education as requirement for fulfillment of master's degree. We were provided with 452 different theses in Nepali language which we will be used for annotation.

The screenshot displays the 'Nepali Language Education' collection page on a DSpace platform. The page features a navigation bar with 'Home', 'Browse', and 'Help' links, a search bar, and a 'Sign on to:' link. The main content area is titled 'Nepali Language Education' and includes a 'Browse' section with filters for 'Issue Date', 'Author', 'Title', 'Subject', 'Submit Date', 'Institute Name', 'Level', and 'Country'. Below the filters is a 'Subscribe' button and a 'Collection's Items' section showing a list of theses sorted by 'Submit Date in Descending order'. The list includes three theses with their respective details. On the right side, there is a 'Discover' sidebar with a list of authors and their associated theses.

Preview	Issue Date	Title	Author(s)	Institute Name	Level	Country
	2015	राजधानी दैनिक पत्रिकाको सम्पादकीयको अध्ययन (RajDhani Dainik Patrikako Sampadakiyako Adhyayan)	काफ्ले Kafil, योगेश Yogesh	Central Department of Education	Masters	-
	2014	छ कक्षाका विद्यार्थीहरूको नेपाली भाषा शिक्षणमा निरन्तर मूल्यांकनको अध्ययन (Chha Kakshaka Bidhyarthiharuko Nepali Bhasa Shikshan ma Nirantar Mulyankan ko Adhyayan)	घले Ghale, जनी बहादुर Jani Bahadur	-	Masters	-
	2010	माध्यमिक तह(कक्षा १०) तथा उच्च माध्यमिक तह(कक्षा ११)का अनिवार्य नेपाली भाषा पाठ्यक्रमको तुलनात्मक अध्ययन (Madhyamik Tah (Kaksha 10) तथा Uchcha Madhyamik Tah (Kaksha 11) ka Anivarya Nepali)	धमला Dhamala, अनिरुद्र Anirudra	Hetauda Campus, Hetauda	Masters	-

Figure 3-1: Collection of Nepali theses



## 3.2 Dataset Preparation

Due to the lack of a readily available dataset, we had to take some measure to prepare accurate and acceptable dataset like what is to be determined as plagiarism and plagiarism free texts like general knowledge and observed facts.

### 3.2.1 Standard Rules Regarding Citation

Although you should use sources creatively and flexibly to help you generate ideas and sharpen your argument, there are some hard-and-fast rules about the way sources should be acknowledged in your project.

- When you quote two or more words verbatim, or even one word if it is used in a way that is unique to the source.
- When you introduce facts that you have found in a source.
- When you paraphrase or summarize ideas, interpretations, or conclusions that you find in a source.
- When you introduce information that is not common knowledge or that may be considered common knowledge in your field, but the reader may not know it.
- When you borrow the plan or structure of a larger section of a source's argument (for example, using a theory from a source and analyzing the same three case studies that the source uses).
- When you build on another's method found either in a source or from collaborative work in a lab.
- When you collaborate with others in producing knowledge. [12]

There are certain things that do not need documentation or credit, including:

- When you are writing up your own results obtained through lab or field experiments.
- When you are using "common knowledge," things like folklore, common sense observations, myths, urban legends, and historical events (but not historical documents)
- When you are using generally accepted facts (e.g., pollution is bad for the environment) including facts that are accepted within particular discourse

communities (e.g., in the field of composition studies, "writing is a process" is a generally accepted fact). [13]

### **3.3 Dataset Analysis**

The dataset of following type of plagiarism will be prepared for training the model using Support Vector Machine.

#### **3.3.1 Complete Plagiarism**

It is essentially copying the entirety of someone's work and labelling it as your own. It is equivalent to intellectual theft and stealing. This will be the most easily detectable type of plagiarism by the system as there will be no changes from the original work. We will be preparing 500 of such paragraphs with complete plagiarism.

Original Text:

फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ गते बडादसैंको महानवमीका दिन पिता भुवनराज भट्टराई र माता दिव्य कुमारीको कोखबाट काठमाडौंको पकनाजोल सल्लाघारीमा भएको हो ।

Plagiarized Text:

फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ गते बडादसैंको महानवमीका दिन पिता भुवनराज भट्टराई र माता दिव्य कुमारीको कोखबाट काठमाडौंको पकनाजोल सल्लाघारीमा भएको हो ।

#### **3.3.2 Direct Plagiarism**

Direct or verbatim plagiarism occurs when an author copies the text of another author, word for word, without the use of quotation marks or attribution, thus passing it as his or her own. In that way, it is like complete plagiarism, but it refers to sections rather than all of texts. This type of plagiarism is considered dishonest and it calls for academic disciplinary actions. We will be preparing 500 of paragraphs with direct plagiarism.

Original text:

फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ गते बडादसैंको महानवमीका दिन पिता भुवनराज भट्टराई र माता दिव्य कुमारीको कोखबाट काठमाडौंको पकनाजोल सल्लाघारीमा भएको हो । थर भट्टराई हटाएर खेताला' उपनाम लेख्ने गरेका फणीन्द्रराज खेतालाले साहित्य, शिक्षा, समाज सेवाका क्षेत्रमा उल्लेख्य योगदान दिएका छन् ।

Plagiarized text:

फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ गते बडादसैंको महानवमीका दिन पिता भुवनराज भट्टराई र माता दिव्य कुमारीको कोखबाट भएको हो | राणातन्त्रको विरोध गर्दै राजनीतिमा भाग लिन पुगेका खेतालाले काराबासको सजाय समेत झेलेको पाइन्छ ।

### 3.3.3 Paraphrasing Plagiarism

This is, as published on Wiley, the most common type of plagiarism. It involves the use of someone else's writing with some minor changes in the sentences and using it as one's own. Even if the words differ, the original idea remains the same and plagiarism occurs. Because students often do not have a clear understanding of what constitutes plagiarism, this is most common in academics. This type of plagiarism detection requires analysis of complete sentiment of the text this would be most difficult to recognize and complexity will be high. We will be preparing 1000 different paragraphs with paraphrased texts to feed in the model.

Original text:

फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ गते बडादसैंको महानवमीका दिन पिता भुवनराज भट्टराई र माता दिव्य कुमारीको कोखबाट काठमाडौंको पकनाजोल सल्लाघारीमा भएको हो । थर भट्टराई हटाएर खेताला' उपनाम लेख्ने गरेका फणीन्द्रराज खेतालाले साहित्य, शिक्षा, समाज सेवाका क्षेत्रमा उल्लेख्य योगदान दिएका छन् ।

Plagiarized text:

फनिन्द्रराज खेतलाको जन्म १३ अक्टोबर १९७९ मा काठमाडौंको सल्लाघारीमा भएको थियो । उनी भुवनराज भट्टराई र दिव्या कुमारीका छोरा हुन् । खेतलाले सन् १९९४ मा आफ्नो थर परिवर्तन गरेर खेताला राखे र त्यसपछि उनले थुप्रै कविता र उपन्यास प्रकाशित गरिसकेका छन्।

### 3.3.4 Non-Plagiarized texts

SVM can still perform well with imbalanced class proportions but having balanced class proportions can lead to more reliable and accurate results so we will be preparing 1500 of such non plagiarized texts to feed into the model for training.

Original text:

फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ गते बडादसैंको महानवमीका दिन पिता भुवनराज भट्टराई र माता दिव्य कुमारीको कोखबाट काठमाडौंको पकनाजोल सल्लाघारीमा भएको हो ।

Plagiarism free text:

खेतालाका सबै कृतिका बारेमा त्यति समीक्षा र टिप्पणी भएको नदेखिए पनि केही रचना र कृतिहरूका बारेमा उल्लेख भएका टिप्पणी र तिनका समीक्षालाई यहाँ प्रस्तुत गरिएको छ ।

Table 3-1: Number of Datasets

Types of Texts	Number of Paragraphs
Complete Plagiarism	500
Direct Plagiarism	500
Paraphrasing Plagiarism	1000
Non-Plagiarized	2000

Total = 4000

## **4. METHODOLOGY**

### **4.1 NLP Approach**

Natural Languages Processing refers to processing of human understandable language by the machines. NLP involves ability to understand text and spoken words by the computers. Only few NLP works in Nepali language has been performed mostly due to the increased complexity and lack of available resources which isn't sufficient for our work.

#### **4.1.1 Corpora of Plagiarized Texts**

In this project we mainly focus on detection of external plagiarism i.e., when both source text and suspicious text are present. The data set is manually prepare following the guidelines for citations as discussed in above section.

### **4.2 Procedural Sequence**

#### **4.2.1 Data Collection and Preprocessing**

The corpus of text with fake plagiarized texts has been classified into four categories of plagiarism in the data set provided from Kaggle. The general preprocessing techniques used on the text are

- Parsing Paragraphs into words: The text in the document is split into sentences and then into words and thereby allowing the individual words to be treated as vectors quantity or tokens
- Removal of Punctuation: The unnecessary punctuation symbols which give no meaning to the texts are removed.
- Parts of Speech tagging: Different tokens are assigned with their own grammar tags according to the parts of speech like noun, verb, adverbs etc.
- Stemming and Lemmatization: The individual words are transformed into their stems in order to generalize the comparison analysis. The different forms of words are normalized into their general form. Example riding, rides, ridden, etc. are normalized into ride. In lemmatization the words like better best are normalized into good.

- **Similarity Adjustment:** The semantic analysis of the text is done by measuring the similarities of the words in the text and looking for the synonyms for the words considering the POS tag

#### 4.2.2 Model Building and Machine Training

The text in the corpus dataset is divided into near copy, light revision, heavy revision and non-plagiarism categories. Plagiarism detection is based on lexical analysis of the tokens made from parsing of the text. The tokens of the suspicious documents are compared with the tokens of original document. Overlap Coefficient is used to compare the tokens of the suspicious and the original documents. The matching process takes the normalized word with the same POS. The value of each text is compared to the value of the original text and system are classified into the four categories. The greater the value of the coefficient defines the greater in similarity between the suspicious document and the original document.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad \dots\dots\dots 4.1$$

The Jaccard Similarity is a very powerful analysis technique, but it has a major drawback when the two sets being compared have different sizes. Consider two sets, A and B, where both sets contain 100 elements. Now assume that 50 of those elements are common across the two sets. The Jaccard Similarity is  $J(A, B) = 50 / (100 + 100 - 50) = 0.33$ . Now if we increase set A by 10 elements and decrease set B by the same amount, all while maintaining 50 elements in common, the Jaccard Similarity remains the same. And there is where Jaccard fails: it has no sensitivity to the sizes of the sets. [14]

The Overlap Coefficient, also known as the Szymkiewicz–Simpson coefficient, is defined as the size of the intersection of set A and set B over the size of the smaller set between A and B.

$$OC(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)} \quad \dots\dots\dots 4.2$$

In Machine Learning, the index is able to quantify the similarities between computer's identified texts and the training data sets. The corpus of the data is to be used as testing data for the machine training. The support vector machine usually deals with pattern classification that means this algorithm is used mostly for classifying the different types of patterns. the main idea behind SVM is the construction of an optimal hyper plane, which can be used for classification, for linearly separable patterns. The optimal hyper plane is a hyper plane selected from the set of hyper planes for classifying patterns that maximizes the margin of the hyper plane i.e., the distance from the hyper plane to the nearest point of each pattern.

The equation shown below is the hyper plane representation:

$$aX + bY = C \quad \dots\dots\dots 4.3$$

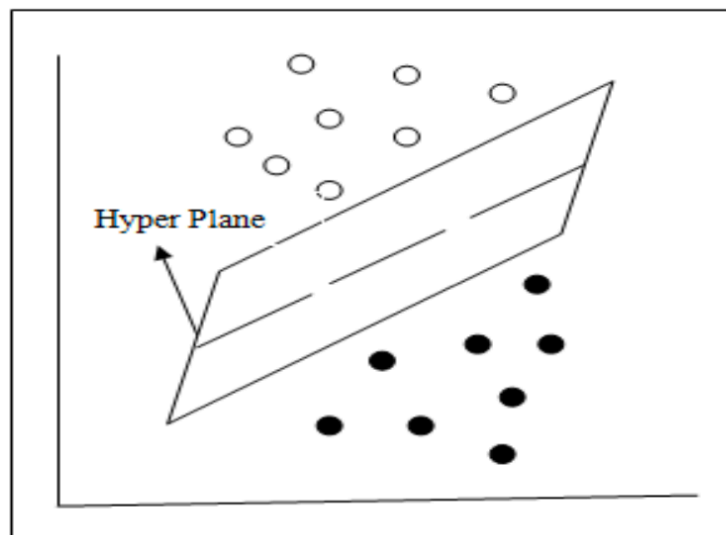


Figure 4-1: Hyperplane [15]

The figure above is the basic idea of the hyper plane describing how it looks like when two different patterns are separated using a hyper plane, in a three dimension. Basically, this plane comprises of three lines that separates two different in 3-D space, mainly marginal line and two other lines on either side of marginal lines where support vectors are located.

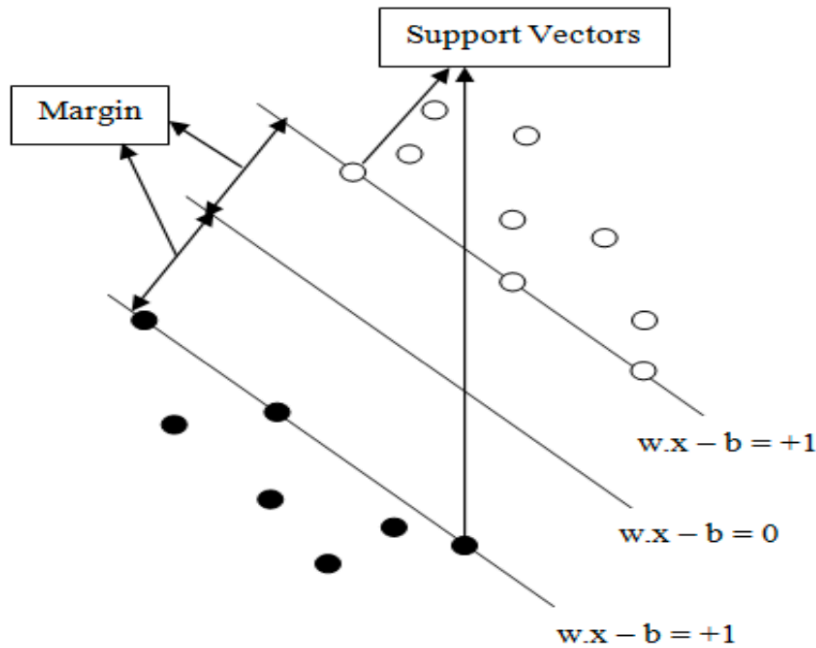


Figure 4-2: SVM model [15]

The figure 4-2 is the simple model for representing support vector machine technique. The model consists of two different patterns and the goal of SVM is to separate these two patterns. The model consists of three different lines. The line  $w \cdot x - b = 0$  is known as margin of separation or marginal line. The lines  $w \cdot x - b = 1$  and  $w \cdot x - b = -1$  are the lines on the either side of the line of margin. These three lines together construct the hyper plane that separates the given patterns and the pattern that lies on the edges of the hyper plane is called support vectors. The perpendicular distance between the line of margin and the edges of hyper plane is known as margin. One of the objectives of SVM for accurate classification is to maximize this margin for better classification. The larger the value of margin or the perpendicular distance, the better is the classification process and hence minimizing the occurrence of error. [3] The main objective of SVM is to maximize the margin so that it can correctly classify the given patterns i.e., larger the margin size more correctly it classifies the patterns.

### 4.3 Use Case Diagram

The use case diagram for the plagiarism detection system represents the interactions between the actors and the system through use cases. The use cases depict the different functionalities of the system such as paper submission by the student, review of the



plagiarism report and database management by the professor; management of database and account information by the admin, comparison of sources to detect plagiarism, and generation of the plagiarism report. The "include" relationship is used to show that a use case depends on another use case and is included as a part of it. The compare Sources use case is initiated by the system as a result of the Submit Paper use case. The Generate Report use case is initiated by the system as a result of the Compare Sources use case. The Generate Report use case is initiated by the system as a result of the Compare Sources use case.

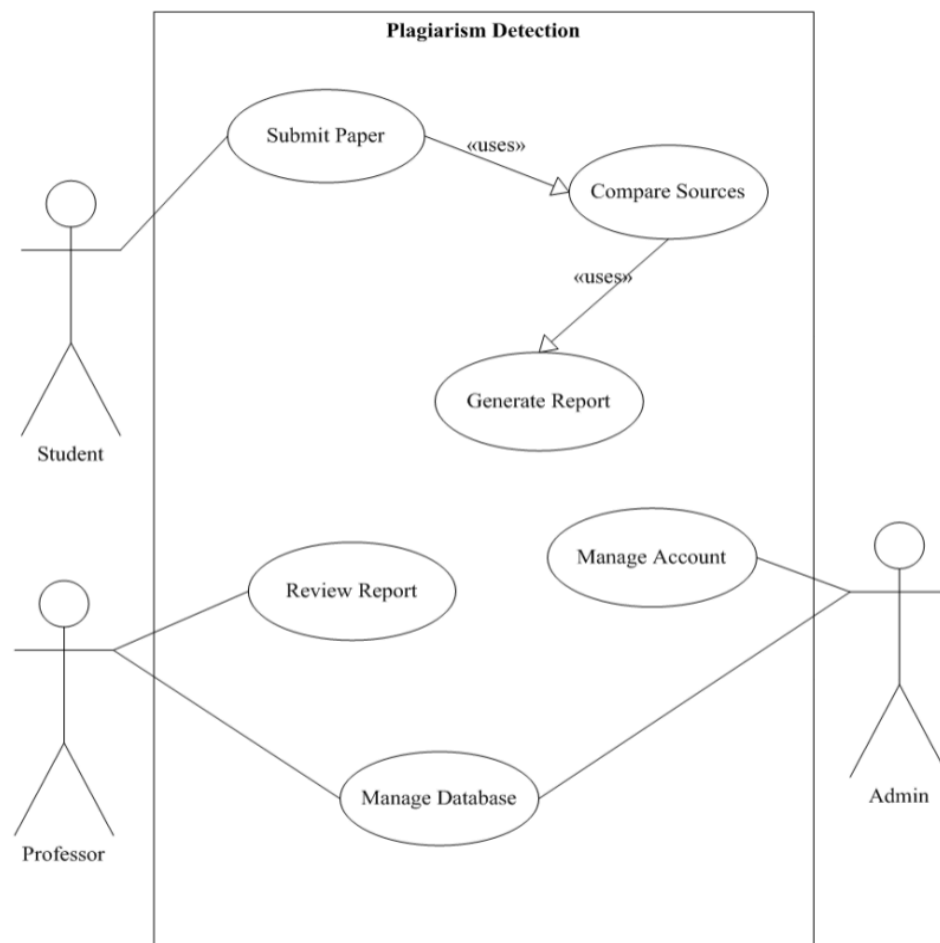


Figure 4-3: Use Case Diagram

#### 4.4 System Architecture

The figure below describes the architecture of the proposed system. The application will provide user to insert the documents into the system one original and other suspicious ones then preprocesses both original and suspicious theses. Next the

application will estimate the similarities between the documents provided and provide user with the plagiarism status through different feature extraction methods.

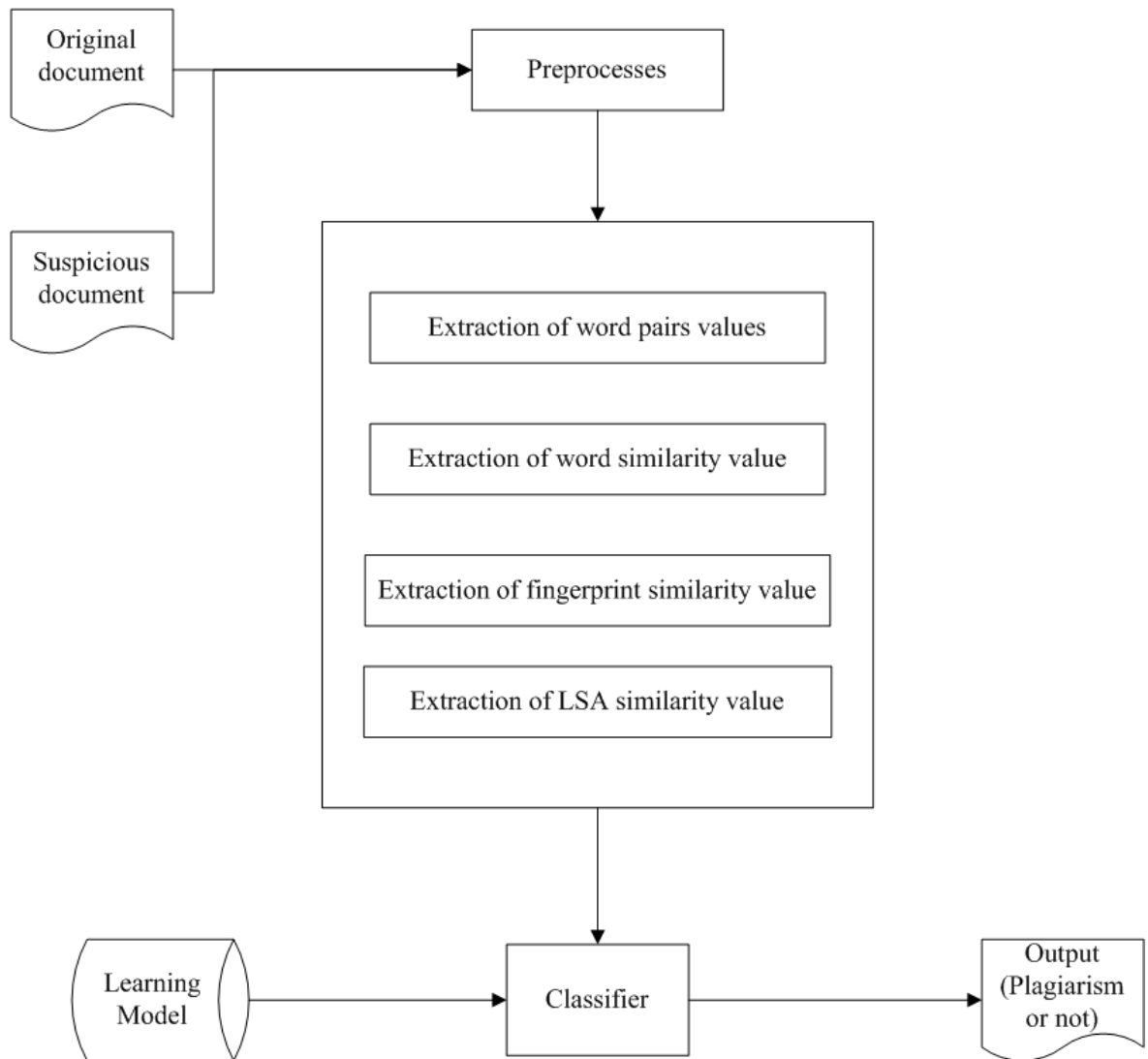


Figure 4-4: System Architecture

#### 4.4.1 Feature Extraction

- **Word Pair:** We chose word pair as a feature to be extracted in creating model and performing classification because the higher the count of consecutive similar words in the text the more likely the text to be plagiarized. Word Pair feature is expected to give information of different words having similar

context. We list the possible words having similar meaning and context in the training data to generate such word pairs.

- **Word Similarity:** Words similarity is chosen to be one of the features because plagiarism sentence tends to have a high-level similarity with the source sentence. But to decide plagiarism or not by looking only at the words similarities is not guaranteed to be always right. For example, the value of this attribute for sample sentence is 1, because to-be-detected and source sentences have 100% words similarity.
- **Fingerprint Similarity:** Fingerprint similarity is chosen because it provides information of similarity between sentences in a more detailed level, which is in the level of n-gram structure. The similarity is the percentage of similar fingerprint between sentences. Fingerprint gives a more detailed similarity value (the n-gram) rather than words similarity.
- **LSA Similarity:** LSA similarity is chosen because it provides information in the form of conceptual similarity of context (semantics) between the two sentences. Conceptual similarity of context has a great influence in determining plagiarism between sentences. We calculate the cosine similarity between to-be-detected sentence semantic vector and source sentence semantic vector. The result is 1, so the value of this attribute is 1.

Support Vector Machine (SVM) is learning algorithm that analyze data and recognize patterns. It makes a hyperplane that can separate two classes. SVM models the existing data into points in a space. The location of each point depends on the value of the features used.

#### 4.5 Flowchart of Proposed System

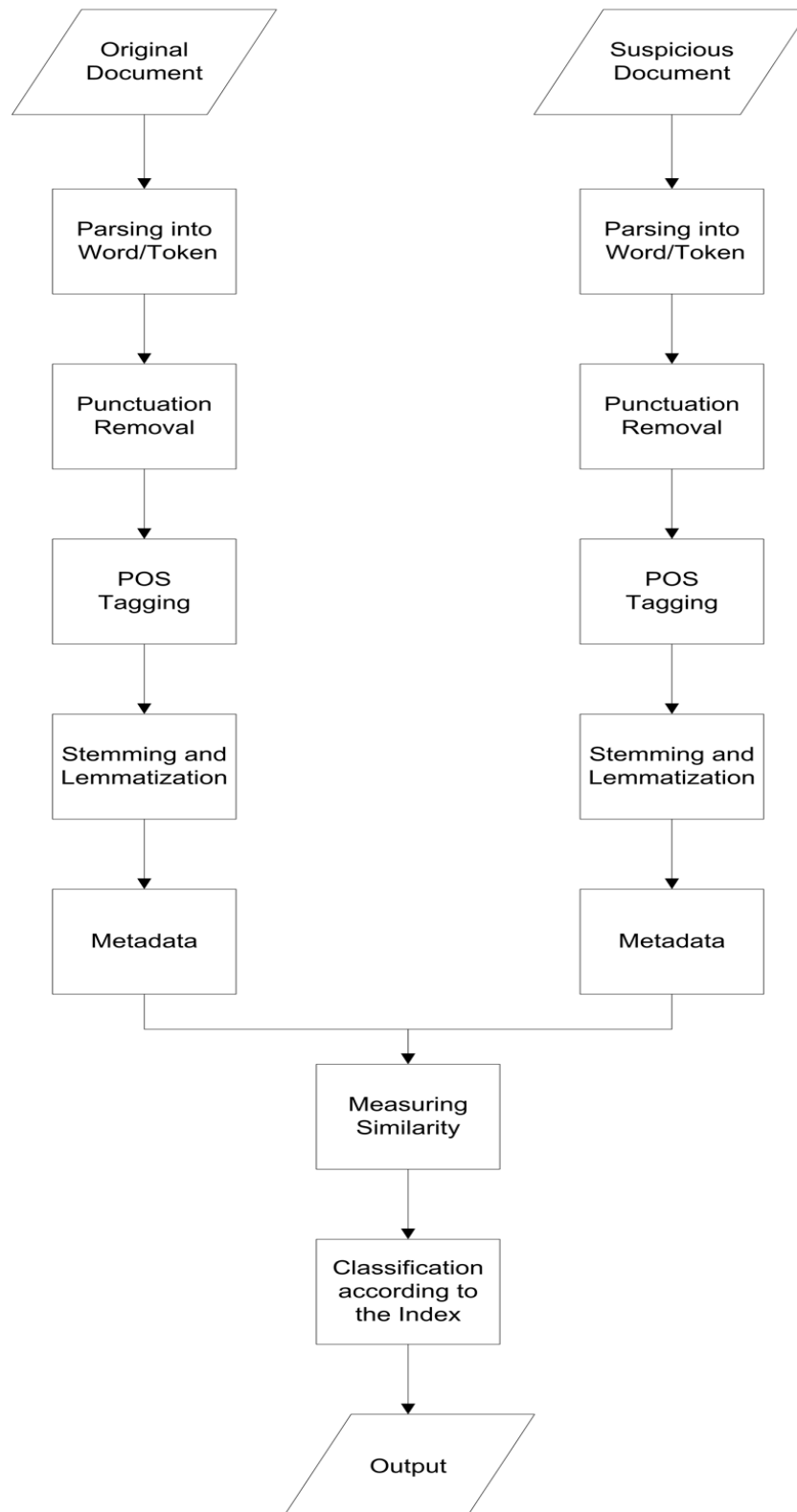


Figure 4-5: Flowchart of Proposed System

## 4.6 Instrumentation Tools.

### 4.6.1 Programming Language

- **Python:** Python Programming Language is mainly used to implement Natural Language Processing for better analyzing and machine training for our model. It provides us with various packages to implement simple mathematical formulae like NumPy, matplotlib to complex data science packages like Scikit-learn and NLKT packages.
- **NKLT:** Natural Language Toolkit is a toolkit build for working with NLP in Python. It provides us various text processing modules like punctuation removal, stop words removal, tokenization etc.
- **Scikit-learn:** Scikit-learn is a python module for implementing machine learning. It provides us with efficient version of large number of algorithms especially for model training through SVM
- **Pandas:** Pandas is a python library used to data analyzing and manipulation. It let us to draw conclusion from big sets of data.
- **Matplotlib:** It is a graph plotting library of python which helped us to visualize the data frame and select suitable model for SVM training.
- **Django:** Django is an open-source high-level Python web framework. It was used for development of web aspect of our project and connecting the machine learning codes with the web application.

## 5. IMPLEMENTATION DETAILS

### 5.1 Dataset Preparation

The dataset used for analysis in this minimal model of plagiarism detection in English language was collected from the GitHub repository of user "gauravansal" [16]. The original dataset consisted of four levels of plagiarism, which were labeled as cut, non-plagiarized, light plagiarized and heavy plagiarized. These levels were each organized in separate text files. In order to simplify the analysis, we decided to create a new dataset that only included two levels: plagiarized and non-plagiarized. This new dataset was created by combining the information from the original text files into a single csv file with a new label. This allowed us for a more straightforward analysis.

Due to the limited resources available for Nepali language, there weren't any data sets to work on Nepali language plagiarism detection. We had to prepare the dataset from scratch and label it ourself. The first step in this process is to create a paragraph-level dataset to assess plagiarism at the paragraph level. The dataset is designed to include four different types of plagiarism: complete plagiarized texts, direct plagiarized texts, non-plagiarized texts, and paraphrased plagiarized texts. To ensure the quality and consistency of the dataset, we decided to use the standards provided by poorvucenter Yale University [12]. The goal of this dataset creation is to provide a comprehensive resource for Nepali language plagiarism detection and to fill the gap in the limited resources currently available for this language. This multi-level dataset will allow for a thorough evaluation of Nepali language plagiarism and help to increase the performance and accuracy of this application.

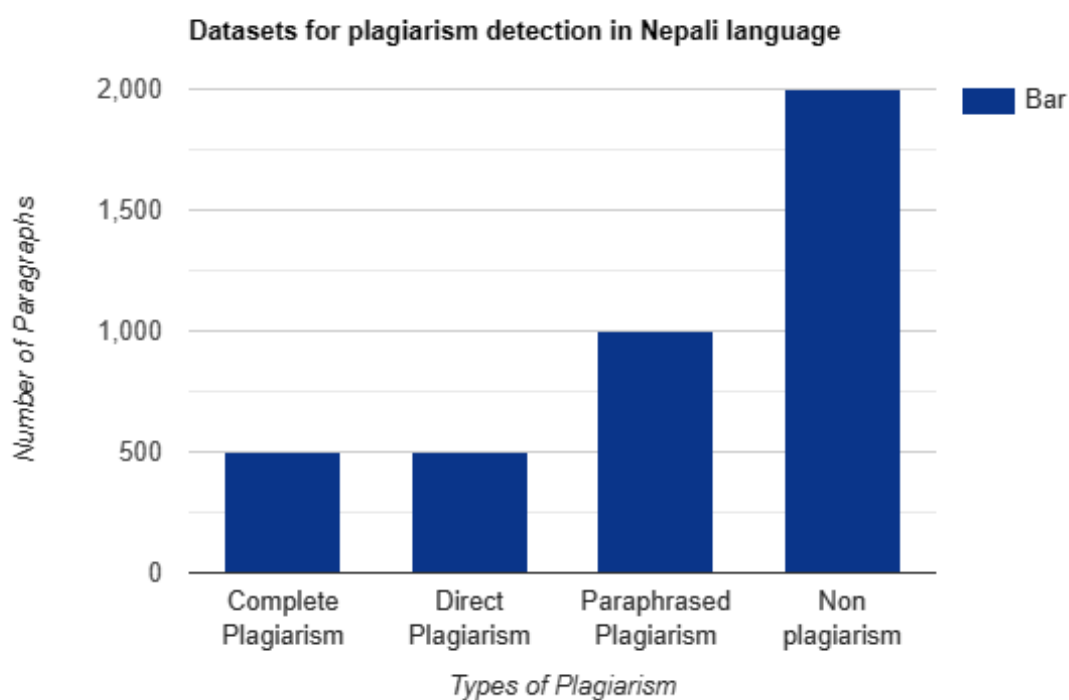


Figure 5-1: Dataset Distribution Chart

## 5.2 Text Preprocessing

Different types of text pre-processing have been applied to the source and suspicious texts to clean and prepare it for the feature extraction.

- **Punctuation removal:** All the punctuation symbols present in the text are replaced with the space using `punc_removal(text)` function from the preprocessing module. This function returns the string after replacing each punctuation with space. The NLTK library provided us with the list of English punctuation to be removed. For Nepali language a additional punctuation ‘|’ needed to be removed.

Input text = “नेपाल धर्मनिरपेक्ष, बहुसांस्कृतिक, बहुभाषिक र बहुधार्मिक राष्ट्र हो । ”

Output text = “नेपाल धर्मनिरपेक्ष बहुसांस्कृतिक बहुभाषिक र बहुधार्मिक राष्ट्र हो”

- **Stop-word removal:** Non-functional words. are removed from the token list as they have no important explanation. This is done using `stopword_removal(token list)` function from the preprocessing module. It returns the cleaned token list.

Some examples of Nepali stopwords are: {अक्सर, अगाडि, अगाडी, अनि, अन्यथा, आदि, उनले, एकदम },etc [17]

- **Tokenization:** Tokenization is the process of dividing the text into the tokens/words. This is done using `tokenization(text)` function from the preprocessing module. This function returns the list of tokens.

Input text= “नेपाल धर्मनिरपेक्ष बहुसांस्कृतिक बहुभाषिक र बहुधार्मिक राष्ट्र हो । “

Output = “ [ 'नेपाल', 'धर्मनिरपेक्ष', 'बहुसांस्कृतिक', 'बहुभाषिक', 'र', 'बहुधार्मिक', 'राष्ट्र', 'हो' ]

- **Lemmatization:** Lemmatization is the process of map words/tokens to a common base/root word.

Words=[ समाचार , संवेदना ]

Root word = [आचार, वेदना]

- **POS tagging:** POS tagging is the processing of assigning the Part of Speech tag to each token.

Text = "१० वर्षीया बालिका बलात्कारपछि हत्या गर्ने सार्वजनिक"

Output: [('१०', 'CD'), ('वर्षीया', 'JJ'), ('बालिका', 'NN'), ('बलात्कारपछि', 'IN'), ('हत्या', 'NN'), ('गर्ने', 'VBNE'), ('सार्वजनिक', 'JJ')]

Here CD, JJ, NN, IN, VBNE represents cardinal digits, adjectives(large), noun singular, preposition, verb past participle respectively.



These preprocessing steps helps to improve the accuracy of plagiarism detection by reducing noise in the text data and making it easier to compare documents. We have used tokenization, punctuation and stop-word removal for calculating bigram, LSA and fingerprint similarity and added lemmatization and pos-tagging for calculating word similarity.

### 5.3 Feature Extraction

- **Word Pair or N-gram:** Here, we divided the text into the bigrams. Bigrams are two token sequences. Thus, created list of bigrams of suspicious and source text are used as a feature to calculate the overlap similarity.
- **Word Similarity:** The dictionary of the similar word is collected and the pos tagging is done to each token in the paragraph. Then, the different tokens of suspicious and original paragraph are compared using dictionary of similar word and considering the POS Tag.
- **Fingerprint Similarity:** Fingerprints are unique identifiers used to distinguish one text from another. In the context of text analysis, fingerprints are used to determine the similarity between two pieces of text. We started by dividing the original text and the suspicious text into n-gram tokens. Next, the algorithm assigns a hash value to each of the n-gram tokens using a min-hash function. The hash values are then used to create a fingerprint of each text. This fingerprint is a compact representation of the text that retains its unique identifying features.
- **LSA Similarity:** LSA (Latent Semantic Analysis) is a method used to determine the similarity between two pieces of text, such as paragraphs. It works by creating a matrix of the frequency of each word (TF-IDF) and then performing a Singular Value Decomposition (SVD) on that matrix. The SVD process results in the identification of latent topics that are present in the text.

These features are used to calculate the different similarity factor between the text using Jaccard and cosine similarity formula.

## **5.4 Model Training and Testing**

Support Vector machine (SVM) model is used to train the dataset of Plagiarism detection in Nepali language. We tried the model on both the paragraph level dataset (95 paragraphs). For now, we are training the model in paragraph level dataset. The dataset was divided into two parts: 80 % for training and 20% for testing.

During the training phase, the SVM model is fed with a labeled dataset of texts, where each text is labeled as either original or plagiarized along with their different similarity values. The SVM model finds the pattern between the suspicious and original paragraph using four features. Hyperplane which maximally separate the two classes is created by studying those patterns.

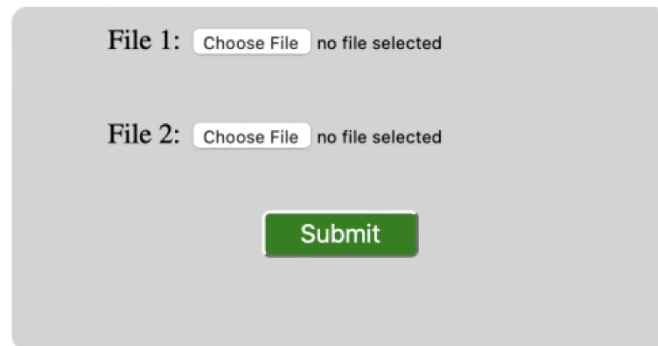
During the testing phase, the SVM model is applied on the testing dataset. The SVM model uses the learned boundary (hyperplane) to predict the label (original or plagiarized) of each test paragraphs. Accuracy of SVM model is calculated using these predictions. To make the model performance satisfactory, we have adjusted the different hyperparameters such as kernel type, regularization parameter, degree.

## **5.5 Model Deployment**

A minimal web application for English plagiarism detection has been created using Django, Python, HTML and CSS and can be run locally on a host machine.

When a suspicious and source text file are submitted to the application, the text files are saved to a MySQL database through the Django model. Then the contents/paragraphs of the text file are accessed and these paragraphs are cleaned using different pre-processing techniques. Feature attraction is done on those cleaned tokenized paragraphs and the similarity value from the feature are fed to the trained SVM model. Then the model predicts the label of suspicious text. Then similarity value of features and the label of suspicious text is shown in the output as shown below:

# Plagiarism Detector



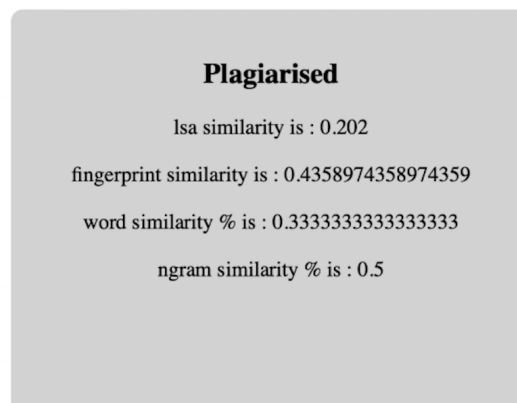
The application interface is a light gray rounded rectangle. It contains two file selection sections. The first section is labeled 'File 1:' and includes a 'Choose File' button and the text 'no file selected'. The second section is labeled 'File 2:' and also includes a 'Choose File' button and the text 'no file selected'. Below these sections is a green 'Submit' button.

File 1:  no file selected

File 2:  no file selected

Figure 5-2: Application Interface

## **Result :**



The result page is a light gray rounded rectangle. It features a bold heading 'Plagiarised' at the top. Below this heading, four lines of text display similarity scores for different methods: 'Isa similarity is : 0.202', 'fingerprint similarity is : 0.4358974358974359', 'word similarity % is : 0.3333333333333333', and 'ngram similarity % is : 0.5'.

**Plagiarised**

Isa similarity is : 0.202

fingerprint similarity is : 0.4358974358974359

word similarity % is : 0.3333333333333333

ngram similarity % is : 0.5

Figure 5-3: Result page

## 6. RESULT AND ANALYSIS

### 6.1 Accuracy with features

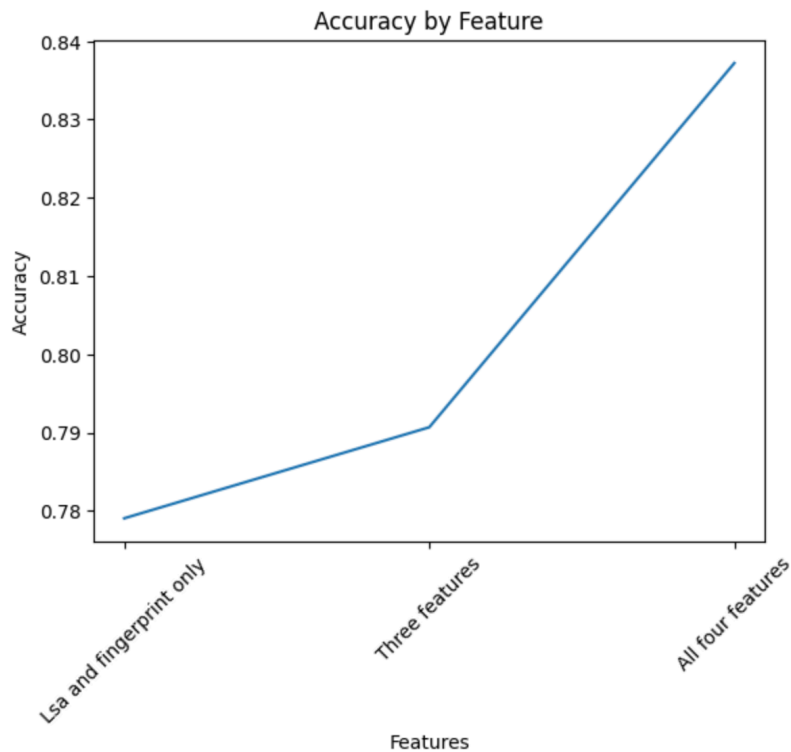


Figure 6-1: Accuracy plot vs Features

This is merely a preliminary sketch of our real project, so a substantial error rate is anticipated. We used a small dataset of about 100 paragraphs only so due to less training data small accuracy in the test data is observed.

It has been observed that all the features play a vital role in improving the accuracy of the model. As in figure above when only LSA and fingerprint similarity were used the accuracy was 78% and when bigram was introduced it increased to 79%. When all four of them LSA, fingerprint, bigram and word similarity were used it increased to 83%.

## **7. REQUIREMENT ANALYSIS**

### **7.1 Feasibility Analysis**

The development of software is disturbed mainly by scarcity of hardware resources, software resources and time constrain. So, it is necessary for us to think about future outcomes while checking the feasibility of the program at the very beginning of the development of the program. The three considerations involved in the feasibility analysis are:

#### **7.1.1 Economic feasibility**

In our project we do not expect any feasibility costs to be spent on as in this program we have only used the open-source resources as they are easily available. Although, to train the model we need a decent laptop/PC. Our own personal computers were enough to train the machine.

#### **7.1.2 Technical feasibility**

Technical feasibility focuses on the existing resources such as hardware, software. It also focuses on the extent to which the available resources can be used. This project will use Natural Language Processing, Machine Learning and Database Management System. Hard part is to gather the dataset, but we can get the dataset online to train the model. So, this project is technically feasible.

#### **7.1.3 Operational feasibility**

Operational feasibility asks if the system will work when developed and installed also, how easily can users use the program/ software. So, keeping that in mind here we are going to deploy the trained model as a web app, which would be easy to use. So, regarding the operational feasibility, it can be operated by user who can use a simple website.

## **8. REMAINING TASKS**

### **8.1 Completion of Dataset Preparation**

The primary challenge in our project has been the preparation of the Nepali dataset for training the machine. We used Supervised learning for higher accuracy but due to limited resources in Nepali language we were unable to find premade dataset to work with. We collected the available theses in Nepali language from Tribhuvan University and started self-annotating them. We have been preparing four different types of datasets:

- Complete Plagiarized text
- Direct Plagiarized texts
- Paraphrased texts
- Non-Plagiarized texts

We have been on our schedule to complete this task by the end of minor project. We have prepared 25% of the total 4000 paragraphs. The remaining 75% will be prepared by the end of semester which will be carried to final year project to perform operations on these datasets.

### **8.2 Lemmatization of Nepali Words**

Stemming often results in a word that is only close to the root word which may not be found in the dictionary. Lemmatization is a more proper process with the use of vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. These lemmas are obtained by applying the suffix and prefix rules repeatedly and using the root word collection for verification. Research in lemmatization had been conducted which was able to obtain the total accuracy of 70.10%. It was due to a smaller number of words in the corpus for training data. [7] We aim to improve this for our system.

### **8.3 POS tagging in Nepali Language**

Part-of-speech (POS) tagging is essential in Natural Language Processing (NLP) and text analysis applications. POS-tagging is a well-researched problem, but there have been limited efforts in fine-grained Nepali POS-tagging. Most work in Nepali POS tagging is limited to coarse-grained tag sets that do not disambiguate morphological features such as gender, number, honorifics, and person encoded within a word. Nepali is a morphologically rich language. It is a complex language and has almost 75% of its general vocabulary comprising derived and inflectional words rather than headwords. [18]

### **8.4 Computation of Word Similarity using Nepali Dictionary**

For the computation of Word Similarity, we will be needed lemmatized Nepali word with POS tagged which will be searched across the Nepali dictionary with similar POS tag to find the plagiarism using semantic analysis. This feature has been found to be very efficient in plagiarism detection and will increase our accuracy. Words similarity is chosen to be one of the features because plagiarism sentence tends to have a high-level similarity with the source sentence. But to decide plagiarism or not by looking only at the words similarities is not guaranteed to be always right. For example, the value of this attribute for sample sentence is 1, because to-be-detected and source sentences have 100% words similarity.

## 9. APPENDICES

### Appendix A: Project Timeline

Table 9-1: Gantt chart

	Dec 1	Dec 15	Jan 1	Jan 15	Feb 1	Feb 15	Feb 28
Research							
Data Collection							
Dataset Preparation							
English Prototype							
Model Implementation							
UI Development							
Prototype Development							
Testing and debugging							
Documentation							

### Appendix B: Nepali Dataset Snippets

	Original	Additional	Label
0	फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ ...	फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ ...	1
1	फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ ...	फनिन्द्रराज खेतलाको जन्म १३ अक्टोबर १९७९ मा का...	1
2	फणीन्द्रराज खेतालाको जन्म वि.सं. १९७९ असोज १३ ...	प्रस्तुत शोधपत्रको शीर्षक फणीन्द्रराज खेतालाको...	0
3	(क) खेतालाको जीवनी के-कस्तो रहेको छ ? (ख) खेता...	(क) खेतालाको जीवनी के-कस्तो रहेको छ ? (ख) खेता...	1
4	(क) खेतालाको जीवनी के-कस्तो रहेको छ ? (ख) खेता...	प्रस्तुत शोधकार्य निम्नलिखित उद्देश्यमा केन्द...	0



## Appendix C: Code Snippets

```
def punc_removal(text):
    # changing punctuation into white space
    for punctuation in eng_punctuations:
        text = text.replace(punctuation, ' ')
    return text

def tokenization(text):
    #now tokenizing
    token = text.strip().split()
    return token

def stopword_removal(token):
    #removing the stop words
    new_tokens = list() #new_tokens hold the list of words after removing stopwords
    for token in token:
        if token not in stopwords_list:
            new_tokens.append(token)

    return new_tokens
```

```
import joblib

def predict_lab(lsaval,fingval,word_sim,ngram_sim):
    model = joblib.load('saved_model1.pkl')
    new_obs = [[lsaval,fingval,word_sim,ngram_sim]]
    label = model.predict(new_obs)
    return label
```

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler

cell_df = pd.read_csv('final.csv')

feature_df = cell_df[['bigram_similarity','simword_similarity' , 'fingerprint_similarity','lsa_sim']]

#independent variables
x= np.asarray(feature_df)

#dependent variables
y= np.asarray(cell_df['label'])

sc_X = StandardScaler()
x = sc_X.fit_transform(x)

from sklearn import svm

#SVC = support vector classifier
classifier = svm.SVC(kernel='linear', gamma = 'auto', C = 1)

#fitting the model
classifier.fit(x,y)

#predicting label using 4 feature
y_pred = classifier.predict([[0.7,0.4,0.5,0.6]])

```

## References

- [1] D. Harper, "Online Etymology Website," 2 august 2020. [Online]. Available: [www.etymonline.com](http://www.etymonline.com). [Accessed January 2023].
- [2] SAS Institute Inc., "Natural Language Processing (NLP): What it is and why it matters," Analytics Software & Solutions(SAS), [Online]. Available: [https://www.sas.com/en\\_us/insights/analytics/what-is-natural-language-processing-nlp.html](https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html). [Accessed January 2023].
- [3] A. Pradhan, "Support Vector Machine-A survey," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 8, August 2012.
- [4] D. D. A. A.-N. S. R. J. K. L. K. J. G.-D. Ö. Ç. a. D. W.-W. T. Foltýnek, "Testing of support tools for plagiarism detection," *International Journal of Educational Technology in Higher Education*, vol. 17, no. 1, 2020.
- [5] R. K. B. a. A. K. Timalina, "Plagiarism detection framework using Monte Carlo based Artificial Neural Network for Nepali language," in *IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, 2018.
- [6] B. K. Bal, "A novel rule-based recursive stemming algorithm for Nepali Plagiarism Detection".
- [7] C. Sitaula, "A Hybrid Algorithm for Stemming of Nepali Text," Kathmandu, 2013.
- [8] B. K. Bal, "Structure of Nepali Grammar," Kathmandu University, 2004.
- [9] "Detailed Analysis of Extrinsic Plagiarism Detection System Using Machine Learning Approach".

- [10] Z. Ceska, "Plagiarism detection based on singular value decomposition," *Advances in Natural Language Processing*, pp. 108-119, 2008.
- [11] O. i. c. o. R. C. f. E. i. a. development, Interviewee, *Plagiarism Detection in Nepali Journals*. [Interview]. 27 January 2023.
- [12] "When you must Cite," [Online]. Available: <https://poorvucenter.yale.edu/undergraduates/using-sources/understanding-and-avoiding-plagiarism/warning-when-you-must-cite>. [Accessed 2023].
- [13] "Plagiarism Overview," Purdue University, [Online]. Available: [https://owl.purdue.edu/owl/avoiding\\_plagiarism/index.html](https://owl.purdue.edu/owl/avoiding_plagiarism/index.html). [Accessed January 2023].
- [14] B. Rees, "Similarity in graphs: Jaccard versus the Overlap Coefficient," 3 May 2019. [Online]. Available: <https://medium.com/rapids-ai/similarity-in-graphs-jaccard-versus-the-overlap-coefficient-610e083b877d>. [Accessed January 2023].
- [15] C. J. BURGESS, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, June 1998.
- [16] G. Ansal, "Plagiarism Detection Corpus".
- [17] K. Paudyal, *Nepali Stopwords*.
- [18] B. K. B. E. Al., "Nepali Spellchecker". *Centre for Research in Urdu Language Proc.*