Tribhuvan University
Institute of Science and Technology

Foreign Word Extraction in Nepali Texts

Dissertation
Submitted to
Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements
for the Master's Degree in Computer Science and Information Technology

by
Diksha Khadka

November, 2014

Tribhuvan University
Institute of Science and Technology

Foreign Word Extraction in Nepali Texts

Dissertation

Submitted to

Central Department of Computer Science and Information Technology

Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements

for the Master's Degree in Computer Science and Information Technology

by

Diksha Khadka

November 12, 2014

Supervisor

Prof. Dr. Shashidhar Ram Joshi

Co-supervisor

Mr. Tej Bahadur Shahi

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

… … … … … … …

Diksha Khadka
November 12, 2014

Tribhuvan University
Institute of Science and Technology

Central Department of Computer Science and Information Technology

Supervisors's Recommendation

I hereby recommend that this dissertation prepared under my supervision by **Ms. Diksha Khadka entitled "Foreign Word Extraction for Nepali Texts"** be accepted as partial fulfillment of the requirements for the degree of M. Sc. in Computer Science and Information Technology. In our best knowledge this is an original work in computer science.

..…………………………….

……………………………..

Prof. Dr. Shashidhar Ram Joshi
Institute of Engineering (IOE)
Tribhuvan University
Pulchowk, Lalitpur, Nepal

(Supervisor)

Mr. Tej Bahadur Shahi
Lecturer
Central Department of Computer Science and Information Technology (CDCSIT)
Tribhuvan University
Kritipur, Kathmandu, Nepal
(Co-Supervisor)

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master's Degree in Computer Science and Information Technology.

Evaluation Committee

………………………………

Asst. Prof. Nawaraj Paudel
Head of Department (HOD)
Central Department of Computer Science and
Information Technology (CDCSIT)
Tribhuvan University, Nepal

..……………………………..

Prof. Dr. Shashidhar Ram Joshi
Institute of Engineering (IOE)
Tribhuvan University (TU)
Pulchowk, Lalitpur, Nepal

(Supervisor)

……………………………

(External Examiner)

……………………………

(Internal Examiner)

Date : December 16, 2014

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge the contribution of a large number of individuals to this work. I deeply extend my heartly acknowledgement to my respected teacher and dissertation advisor Prof. Dr. Shashidhar Ram Joshi, Head of Department, (IOE), for giving me an opportunity to work under his supervision and for providing me guidance and support throughout this work. With this regard, I wish to extend my sincere appreciation to respected co-supervisor Mr. TejBahadurShahi, who shared his immense knowledge and precious time with me. I want to thank for his guidance, advice, his enthusiasm and patience throughout my dissertation.

I wish to extend my sincere appreciation to respected Head of the Central Department of Computer Science and Information Technology, Asst. Prof. Mr. Nawaraj Paudel for his kind help, encouragement and constructive suggestions.

I would like to express my gratitude to the respected teachers Assoc. Prof. Dr. Tanka NathDhamala, Prof. Dr. SubarnaShakya, Prof. SudarshanKaranjeet, Mr. JagdishBhatta, Mr. Arjun Saud, Mr.LalitaSthapit, Mr. SarbinSayami and others staffs of CDCSIT for granting me broad knowledge and inspirations within the time period of two years.

This dissertation would not have been possible without the advice and support of my friends. Therefore, I would like to express my special thanks to Mr. Dadhi Ram Ghimire, Mr. ShailaMitraNeupane and Mr. SandipAryal. I cannot remain without admiring the efforts put by my husband Mr. BirodhKattel for his exceptional participation on this work.

A big thank you to my whole family, for being by my side, supporting me in every step I take and for being ready to help me in any occasion. My biggest thank you goes to my father Dr.Sarba Raj Khadka and mother Ram Devi Khadka without whom I would have never been able to be where I am today both personally and professionally. This work is dedicated to my parents and my daughter BidishaKattel.

Thank you all for always believing on me.

ABSTRACT

In Nepali text, foreign words, which are mostly transliterations of English words, are frequently used. Foreign words are usually very important index terms in information retrieval since most of them are technical terms or names. So, accurate foreign word extraction is important for high performance of information retrieval. In this study we present a foreign word extraction method for Nepali text document. In order to accurately extract the foreign words, we developed a framework using rule based syllabification.

The performance analysis includes different components such as known words, unknown words and size of training data. The present study of supervised rule based syllabification approach is limited due to the existence of same syllable structure for both Nepali and English words and it use a small dictionary which affects its performance.

During this study, the efficacy has taken over 12000 syllabified words taken from different daily online news sites. The analysis is done taking into account the various factors like Precision and Recall.

In this dissertation, we present a syllabification algorithm for Nepali language. The process of syllabification performs the task of identifying syllables in a word. The correct syllabification rules and algorithms are mainly used in text-to-speech system to improve naturalness of the synthesized speech. We propose an algorithm based on syllable rules matching. The syllable rules matching achieved precision of 83% and recall of 63%.

Table of Contents

1. INTRODUCTION

2. BACKGROUND AND PROBLEM DEFINITION

3. DATA COLLECTION AND IMPLEMENTATION

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
POS	Part-Of-Speech
FW	Foreign Word
HMM	Hidden Markov Model
NLP	Natural Language Processing
NLG	Natural Language Generation
NN	Neural Network
TTS	Text to Speech
SSP	Sonority Sequencing Principle
SVM	Support Vector Machine
IR	Information Retrieval

CHAPTER 1

INTRODUCTION AND PROBLEM DEFINITION

1. 1 Introduction

The use of foreign words in Nepali texts is growing at a high speed. They are mostly from English language and are usually used as transliterated forms. For example, an English word ‘data’ can be transliterated into Nepali word like "डाटा". This causes errors such as the well-known unknown word problem in a morphological analyzer. A foreign word is usually translated to a local word according to their phonetic similarity in the two languages. Such translated words are referred to as transliterations [1]. For example the English word“SYSTEM” is actually used as foreign word “सिस्टम” in Nepali text. This is not registered in Nepali dictionary but used in our daily activities.

Documents present in the World Wide Web are considered to be one of the most useful sources of information. The use of search engine to retrieve the documents can harvest lots of important information which facilitates information exchange and knowledge sharing.It includesboth global and domestic information [2]. Hence, to better understand by local readers, foreign words are often translated into local languages such as Nepali.

Information retrieval is the activity of getting access to information resources relevant to an information need from a collection of information resources. The term “Information Retrieval (IR)”, refers to the retrieval of unstructured records, that is, records consisting primarily of free-form natural language text.[3] However, IR research has focused on retrieval of natural language text, a reasonable emphasis given the importance and immense volume of textual data, on the internet and in private archives.

Foreign word (FW) which are mostly transliteration of English words, are frequently used and growing at high speeds. This is mainly due to the World Wide Web and internet that enable instant access of new information at the global scale. Foreign words are usually very important index terms in information retrieval since most of them are technical terms or names. So, accurate foreign word extraction is crucial for high performance of information retrieval. The newly introduced foreign words usually remain unregistered in any dictionary during the significant amount of time [4]. This causes the well known, unknown word problem in natural language processing.

In Nepali sentence like “नेपालमाटेक्नोलोजीकोविकासद्रुतगतिमाभइरहेकोछ।”, here the word “टेक्नोलोजी” is not registered in any Nepali dictionary but mostly used in different articles as a Nepali text.

In information retrieval, usually only nouns are indexed, so accurate noun extraction is important. However, the unknown word problem caused by foreign words significantly hinders the noun extraction task since noun extraction in many language like Chinese, Japanese, Korean,etc accompanies word segmentation problem.[5] Most of the foreign words are nouns and also important content carriers so that they are usually first class index term. Hence, accurate foreign word extraction is a very important issue in information retrieval.

Nepali words can be broadly classified into content words (nouns, verb, adjective, etc). In addition, nouns are relatively freely joined together to form compound noun. Foreign words are detected by using supervisedrule based syllabification method.

1.2 Objectives

Given an initial text fragment, the foreign word extraction problem is to identify the foreign words that are mostly the transliteration of English words. The objectives of this work are:

- To build a model to extract the foreign words from Nepali text.

1.3 Problem Statement

Nepali text is composed of various kinds of words. A Nepali word in the text can be written as

$$W = (s_1, s_2, s_3, \ldots\ldots s_n)$$

Where S_i is the i^{th} Nepali syllable in a word W. Here, each word can be of Nepali origin word or foreign word. So, the problem is to classify each word $S_i$into either class Nepali or Foreign.

$$V= (v_1,v_2,v_3\ldots\ldots v_n)$$

is a set of valid Nepali syllable structures, then

If $W - V = \emptyset$, then the word is Nepali otherwise it is foreign.

1.4Organization of Thesis

The rest of the thesis is organized as follows:

In chapter 2, section 2.1 presents background and section 2.2 discusses background concept i.e. phonological structure of Nepali language needed for this study, section 2.4 describes related works that has been done in this area, are included in literature review.

Chapter 3 describes data collection and implementation details in which section 3.1 presents collection of Nepali corpus. Chapter 4 describes testing and analysis done during this study. Finally, Chapter 5 contains conclusion, and further recommendation of this study.

CHAPTER 2

BACKGROUNDAND LITERATURE REVIEW

2.1 Background

2.1.1 Linguistic and Natural Language Processing

Natural Language Processing (NLP) has been developed in 1960 as a subfield of Artificial Intelligence and Linguistics [6]. The aim of NLP is studying problem in the automatic generation and understanding of natural language. A natural language is any of the languages naturally used by humans, i.e not an artificial or machine language such as a programming language like C language, Java, Perl etc.

NLP is a convenient description for all attempts to use computers to process natural language. NLP is also an area of artificial intelligence research that attempts to reproduce the human interpretation of language for computer system processing. The ultimate goal of NLP is to determine a system of language, words, relations, and conceptual information that can be used by computer logic to implement artificial language interpretation. NLP includes anything a computer needs to understand natural language (written or spoken) and also generate the natural language. To build computational natural language systems, we need Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLG systems convert information from computer databases into normal-sounding human language, and NLU system convert samples of human language into more representation that are easier for computer programs to manipulate [6]. Some of the important levels of NLP are as follows:

Phonological Analysis: Phonology is the study of sound system in a language. The minimal unit of sound system is the phoneme which is capable of distinguishing the meaning in the words. The phonemes combine to form a higher level unit called syllable and syllables combine to form words. Therefore, the organization of the sounds in a language exhibits the linguistic as well as computational challenges for its analysis. This level deals with the interpretation of speech sounds within and across words. There are, in fact, three types of rules used in phonological analysis: 1) phonetic rules - for sounds within words; 2) phonemic rules – for variations of pronunciation when words are spoken together, and; 3) prosodic rules – for fluctuation in stress and intonation across a sentence. In an NLP system that accepts spoken input, the sound waves are analyzed are encoded into a digitized signal for interpretation by various rules or by comparison to the particular language model being utilized.

Morphological Analysis: This level deals with the componential nature of words, which are composed of morphemes. – the smallest units of semantic meaning. For example, the word preregistration can be morphologically analyzed into three separate morphemes: the prefix ‘pre’, the root ‘registra’, and the suffix ‘tion’. Since the meaning of each morpheme remains the same across words, humans can break down an unknown word into its constituent morphemes in order to understand its meaning [6]. Similarly, an NLP system can recognize the meaning conveyed by each morpheme in order to gain and represent meaning. For example, adding the suffix – ‘ed’ to a verb conveys that the action of the verb took place in the past. This is a key piece of meaning, and in fact, is frequently only evidenced in a text by the use of the –ed morpheme. Typically, a natural language processor knows how to understand multiple forms of a word i.e. its plural and singular, for example, *ghar(घर)* ‘house’ *ghar-haru (घरहरु)* ‘house-s’. From structural point of view, the words can be simple, complex and compound. For example, *gahr*‘house’, *ghar-haru* ‘house- plural’, *gahr-ghar-ai* ‘each house’.

Lexical Analysis: At this level, humans, as well as NLP systems, interpret the meaning of individual words. Several types of processing contribute to word-level understanding – the first of these being assignment of a single part-of-speech (POS) tag to each word. In this processing, words that can function as more than one part-of-speech are assigned the most probable part-of-speech tag based on the context in which they occur. The lexical level may require a lexicon, and the particular, approach taken by an NLP system will determine whether a lexicon will be utilized, as well as the nature and extent of information that is encoded in the lexicon [6].

Syntactic Analysis: Syntactic analysis uses the results of morphological analysis and lexical analysis to build a structural description of the sentence. The goal of this process, called parsing, is to convert the flat list of words that forms the sentence into a sentence into a structure that defines the units that are represented by that flat list. The important thing here is that a flat list of words has been converted into a hierarchical structure and that the structures correspond to meaning units when semantic analysis is performed [6].

Semantic Analysis: It derives an absolute (dictionary definition) meaning from context; it determines the possible meaning of a sentence in a context. The structures created by the syntactic analyzer are assigned meaning. Thus, a mapping is made between individual words into appropriate objects in the knowledge base or database. It must create the correct structures to correspond to the way the meaning of the individual words combine with each other. The structures for which no such mapping is possible are rejected [6]. Example: the sentence "colorless green ideas…." Would be rejected as it has no such semantic mapping, because colorless and green make no sense.

Pragmatic Analysis: It derives knowledge from external commonsense information; it means understanding the purposeful use of language in situations, particularly those aspects of language world knowledge [6]. Example: If someone says "the door is open" then it is necessary to know which door "the door" refers to; here it is necessary to know what the intention of the speaker: could be a pure statement of fact, could be an explanation of how the cat go in, or could be a request to the person addressed to close the door.

Discourse Integration: The meaning of an individual sentence may depend on the sentences that precede it and may influence the meaning of the sentences that follow it [6]. Example: the meaning of word "it" in the sentence, "you wanted it" depends on the previous discourse integration.

2.1.2 Corpus Linguistics

Corpus linguistics is now seen as the study of linguistic phenomena through large collections of machine-readable texts: corpora. These are used within a number of research areas going from the descriptive study of language learning. Corpus linguistics has developed considerable in the last decades due to the great possibilities offered by the processing of natural language by computers having large storage capacity. The availability of computers and machine-readable text has made it possible to get data quickly and easily and also to have this data presented in a format suitable for analysis. Corpus linguistics is the study and analysis of data obtained from a corpus. The main task of the corpus linguist is not to find the data but to analyze it [7]. Computers are useful, and sometimes indispensable, tools used in this process.

2.2 Phonological Structure of Nepali Language

Nepali is an Indo-Aryan language. It takes its root from Sanskrit, the classical language of India. Nepali was previously known as *khaskura* and the language of the *khasa* kingdom. Nepali is written with the Devanagari alphabet, which developed from the *Brahmi*script [8]. In Nepali language there are 11 vowels and 33 consonants. The script being phonetic in nature, and hence the pronunciation closely resembles the writing system. The script is written from left to right. There is no provision of small and capital letters in the script. The alphabets are written in two separate groups, namely vowels and consonants as shown in the table 1 [8].

Vowels	**अ ,औ ,ओ ,ऐ ,ए ,ऋ ,ऊ ,उ ,ई ,इ ,आ**
Consonants	**क,घ ,ग ,ख , ञ ,द ,थ ,त ,ण ,ढ ,ड ,ठ ,ट ,झ ,ज ,छ ,च , ह ,स ,ष ,श,व ,ल ,र ,य ,म ,भ ,ब ,फ ,प ,न ,धङ**
Vowel signs	◌ौ","◌े","◌ै","◌ृ","◌ु","◌ू","ि◌","◌ी","◌ो","◌ा","◌ँ"

Table 2.1: Alphabets of the Nepali Writing System

The three letters **क्ष**, **त्र** and **ज्ञ** are regarded as special clusters and are dealt with separately from the consonants. The three clusters are formed by the combination of the other consonants with the viram or halanta playing a significant role in the combination [8] as shown below.

क्ष = क +◌्+ ष

त्र = त +◌्+ र

ज्ञ = ज +◌् +ञ

The inventory of Nepali phoneme consists of segmental phonemes being consonants and vowels and super segmental phonemes being tone, juncture and contour, co-occurring with them extra features used in the language. [9]

2.2.1 Introduction to Syllables

A syllable is generally a speech sound of a particular language which can be pronounced with a single puff of breathes. It is an essential element for a morpheme, i.e.: no morpheme can be formed without a syllable. Wikipedia, the free encyclopedia states "Syllables are often considered the phonological 'building blocks' of words." For example, in the word 'Book' [*bʊk*], there is only one syllable, whereas the word 'Copy' [kopi:] is made up of two syllables. We need to break air to pronounce this word between 'co', and 'py'.

The word with a single syllable is called "monosyllabic" (you, go, talk); with two are "disyllabic" (copy, vowel, inbox). In the same way they could be "trisyllabic" or "polysyllabic".

2.2.2 Transliteration

Transliteration is a representation of the words of one language in the script of another, i.e. it is the transcription of one alphabet in another. Some other interesting definitions are :

- The representation of characters or words of one language by corresponding characters of words of another language.
- A systematic way to convert characters in one alphabet or phonetic sounds into another alphabet.
- The translation of text from one writing system into another where the writing conventions of the target writing system are applied. The transliterated text should read naturally in the target script.
- A letter-for letter or sound –for –letter spelling of a word to represent a word in another language.

In multilingual processing, transliteration must be used for handling words in the following categories.

- The names of people, organizations etc. (e.g. Martin (मार्टिन), Microsoft (माइक्रोसफ्ट) etc.)
- Technical or scientific terms, which are used by people working in specific industries. (e.g., computer(कम्प्युटर), pump(पम्प) etc.)
- Foreign language words that are used directly by native speakers rather being translated (e.g. cricket (क्रिकेट), table (टेबल), school(स्कुल) etc.)

- Words about which the language processing system has no information. (Rare words that ar not yet in the lexicon).

Transliteration is also used for the purpose of conveying the pronunciation of certain texts that are in a different language. For instance, books containing prayers and other scriptures (e.g. the Bhagvad Gita and the Upanishads) are often transliterated, which allows people to chant from or quote these texts though they may not be able to read script.

2.2.3 Syllable Tagging

Syllable is a unit which is intermediate between phoneme and word, larger than phoneme and smaller than word [10].So many theories are available in phonetic and phonology to define syllable. In phonetics, the syllables are defined based upon the articulation [10]. But in phonology, the syllables are termed as the different sequences of the phonemes.

Syllabification is the process of dividing a word into its constituent syllables [10]. Syllabification is a TTS (Text to Speech) system is essential for two reasons [10]. First, it helps the implementation of certain letter-to-phoneme rules. Second, syllabification is essential in enhancing the quality of synthetic speech since detecting the syllable will help in modeling duration and improve the synthesized speech intonation.

Oh and Choi (1999) developed more effective foreign word extraction method using the HMM method [11]. They reformulated the foreign word recognition problem as a syllable tagging problem such that each syllable is tagged with the foreign syllable tag [5]. The problem is to determine the most probable tag sequence T given a sentence S:

$$\text{T*} = \underset{T}{\arg\ \max}(S \mid T)P(T) \qquad (2.1)$$

Applying the chain rule and assuming that the current tag depends only on the previous two tags and the current syllable observation depends only on the current tag and the immediate previous tag, the term P (S|T) P (T) may be simplified as follows:

$$P(S \mid T)P(T) = \prod_{i=1}^{n}[p(s_i \mid t_i t_{i-1}) \times P(t_i \mid t_{i-1}t_{i-2})] \qquad (2.2)$$

Where t_0=B (begin of word symbol) and p (t_1|t_0t-$_1$) =p (t_1|t_0) to simplify the notation.

2.2.4 Structure of Syllable

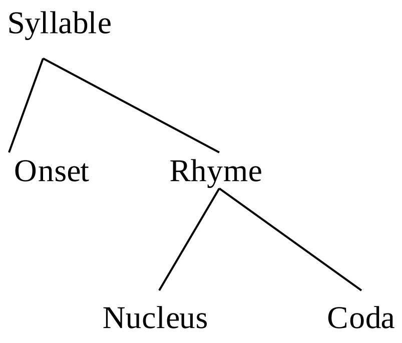


Fig 2.1: Syllable Structure

A syllable generally has two parts: onset and rhyme.

I) **Onset**:

This is initial consonant sound in a syllable. However, it is not necessary that every syllable must have an onset. For example, the word "man" is a monosyllabic word, thus it has only one syllable, i.e.: "man" itself. Here /m/ is the onset. Yet in another monosyllabic word "aunt", there is no onset, since it begins with the vowel.

Sometimes, onset can be not only a single consonant, but a consonant cluster (combination of two or more consonants without vowel in between). For example, in "school", the onset part is "sch" /sk/.

II) **Rhyme**:

It is later part occurring after the onset. Further, rhyme can also be divided into two parts:

Nucleus: It is the essential / obligatory part of syllable, and thus is a must for every syllable of every language. More than often, it is a vowel. For example, in the word 'telephone, there are three syllables /te/, /le/, and /fon/ and each of them contain vowel onsets. They are: /e/, /e/ and /o/.

In some cases, the nucleus can be diphthongs (combined together vowels) or even vowel sequences, i.e. two or more vowels coming together without a consonant in between. E.g: in the word "pair", the nucleus is a diphthong.

Coda: It is the consonant sound at the end of a syllable following nucleus. For example, in "man", /n/ is the coda, and in "school", it is /l/.

Similar to onset, coda is also a non-essential / optional part of the syllable. For example, in the syllable /preI/ from the word "pray" we can't find any coda, but only onset and nucleus. Like onset, consonant clusters can also form coda for a syllable. For example, the coda of "schools" is /ls/, made up of two consonants /l/ and /s/.

The structure of syllable is generally analyzed on the basis of order of vowels and consonants it follows. For example, syllable structure of "school" is CCVC (Consonant-Consonant-Vowel-Consonant).

2.2.5 Nepali Syllable Structure

Like English and most of the languages in the world, in Nepali too, vowels are nucleus of a syllable. So, no syllable can be formed without at least a vowel. In addition, consonant or consonant clusters precede or follow the nucleus vowel as onset or rhyme to form syllables. The word level analysis of Nepali reveals that a word contains at least one syllable and at the most four syllables. Thus mono-syllabic, di-syllabic, tri-syllabic, and a few tetra-syllabic words are found in the language. [9]

Structures of syllables in Nepali language are as follows:

I) **V**

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**
/ə/	c	/əməla:/	Embolicmyrobolon, a fruit
/I/	O	/inar/	Well
/o/	cf]	/obHAno/	Dry

II) **VV**

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**
/əI/	P]	/əIna:/	Mirror
/əU/	cf}	/ə UsaDHi/	Medicine

III) **VC**

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**
/a:TH/	cf7	/a:TH/	Eight

/oTH/	cf]7	/oTH/	Lip
/Ut/	pt\	/UtkaT/	Extreme

IV) <u>CV</u>

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**
/ka:/	sf	/ka:gəz/	Paper
/gHo/	3f]	/gHosada/	Declaration
/tU/	t'	/tUlana/	Comparison

V) <u>CVV</u>

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**
/ləU/	nf}	/laURo/	Stick
/k əI/	s}	/MakəI/	Maize
/DəI/	b}	/ DəIbə/	God
/ra:U/	/fp	/ra:Ute/	An ethnic tribe

VI) <u>CCV</u>

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**
/tjo/	Tof]	/tjo/	That
/bjə/	Ao	/bjaktI/	Individual
/trə/	q	/sUtRa/	Formula

VII) <u>CCVV</u>

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**
/kHja:U/	Vofp	/kHja:Ute/	Extremely thin

VIII) <u>CCCV</u>

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**
/stri:/	:qL	/stri:/	Woman

IX) <u>CVC</u>

Syllable	**Devnagari**	**Exemplary word**	**Meaning of the exemplary word**

/lUk/	n's\	/lUknU/	To hide oneself
/lok/	nf]s\	/a: lok/	Light, Brightness
/bHok/	ef]s	/bHok/	Apettite, Desire

X) **<u>CCVC</u>**

Syllable	Devnagari	Exemplary word	Meaning of the exemplary word
/pwa:l/	Kjfn	/pwa:l/	Hole
/dZHja:l/	‰ofn	/dZHja:l/	Window
/kHja:l/	Vofn	/kHja:l/	Care
/dja:N/	8\ofª\	/dja:NNa/	An onomatopoeia for sound

XI) **<u>CCCVC</u>**

Syllable	Devnagari	Exemplary word	Meaning of the exemplary word
/blja:N/	ANofª\	/blja:Nna/	An onomotpeia for act of slipping

The above study shows that in Nepali too, vowel is a nucleus, essential to form a syllabus. Consonants can occur before and after it. In those examples, it is clear that Nepali allows maximum three consonants together in the initial position of a syllable. But consonant clusters are nowhere seen in the final position in the syllable.

Similarly, regarding vowel sequence, it allows maximum two vowels together, though they are not as much as single vowels.

2.3 Theories of Syllabification

There is some debate as to the exact structure of a syllable. However, phonologists are in general agreement that a syllable consists of a nucleus (vowel sound), preceded by an optional onset and followed by an optional coda [12]. In many languages, both the onset and coda can be complex, i.e., composed of more than one consonant. For example, the word breakfast [break-fast] contains two syllables, of which the first has a complex onset [br], and the second a complex coda [st]. Languages differ with respect to various typological parameters, such as optionality of onsets, admissibility of codas, and the allowed complexity of the syllable constituents. For example, onsets are required in German, while Spanish prohibits complex codas. There are a number of theories of syllabification; we present three of the most important.

2.3.1 Legality principle

The legality principle restricts the segments that can begin and end syllables to those that appear at the beginning and end of words [12]. In other words, a syllable is not allowed to begin with a consonant cluster that is not found at the beginning of some word, or end with a cluster that is not found at the end of some word (Goslin and Frauenfelder, 2001). Thus a word like admit must be syllabified [ad-mit] because [dm] never appears word initially or word finally in English. A short coming of the legality principle is that it does not always imply a unique syllabification.

2.3.2 Sonority Sequencing Principle (SSP)

The Sonority Sequencing Principle (SSP) provides a stricter definition of legality. The sonority of sound is its inherent loudness, holding factors like pitch and duration constant [12]. Low vowels like [a], the most sonorous sounds, are high on the sonority scale, while plosive consonants like [t] are at the bottom. When syllabifying a word, SSP states that sonority should increase from the first phoneme of the onset to the syllable's nucleus, and then fall to the coda (Selkirk, 1984). Consequently, in a word like vintage, we can rule out a syllabification like [vi-ntage] because [n] is more sonorant than [t].

Both the Legality Principle and SSP tell us which onsets and codas are permitted in legal syllables, and which are not. However, neither theory gives us any guidance when deciding between legal onsets.

2.3.3 Maximum Onset Principle

Maximum Onset Principle addresses this by stating we should extend a syllable's onset at the expense of the preceding syllable's coda whenever it is legal to do so [12]. Maximum onset principle is also considered as universal principle of syllabification. If a consonant cluster within a word can be divided into two parts, such that the first is the possible word-final cluster and the second a possible word-initial cluster then a syllable boundary may be placed between these two parts [12]. Ob-struct is the correct word syllabification based on maximum onset principle.

2.4 Literature Review

There has been lot of work done in the field of foreign word detection and extraction. Some of the related works are explained on the following sections.There is a growing body of research on the field of foreign word identification and extraction. The task of identifying transliterated words has been less studied. Stalls and Knight [13] identified the problem- “in Arabic, there are no obvious clues, and it’s difficult to determine even whether to attempt a back-transliteration, to say nothing of computing and accurate one”- but don’t deal with it directly. Oh and Choi [11] studied identification of transliterated foreign words in Korean text, using an HMM on the word syllable structure. They used a corpus of about 1,900 documents in which each syllable was manually tagged as being either Korean or Foreign, and achieved impressive results. However, besides requiring a large amount of human labor, their results are not applicable to Hebrew (or Arabic) as these languages syllables structure is not clearly marked in writing, and even the vowels are not available in most cases. Nwesri*et al*[14] dealt with the identification of transliterated foreign words in Arabic text in the setting of an information retrieval system. They tried several approaches: using an Arabic Lexicon (everything which is not in the lexicon is considered foreign), relying on the pattern system of Arabic morphology, and two statistical n-gram models, the better of which was based on Cavnar and Trenkle’s rank order statistics[13], traditionally used for language identification. For the statistical methods, training was done on manually constructed lists of a few thousands Arabic and foreign words written in Arabic script. They also augmented each of the approaches with hand written heuristic rules.

They achieved mediocre results on their training set, somewhat lower results for their test set, and concluded that the best approach for the identification of transliterated foreign words is the lexicon based method enhanced by hand written heuristics, by which they achieved a precision of 68.4% and recall of 71.1% on their training set, and precision of 47.4% and recall of 57.2% on their test set.

Another related field of research is that of language identification, in which documents are classified according to their language. The problem of finding transliterated foreign words instead of documents or sentences. Algorithms that rely on letter-ngram statistics can be relevant to the foreign words identification task[13]. Two notable works which are based on letter-level ngram statistics. Canvar and Trenkle[13] use rank order statistics and Dunning use Naïve-Bayes classification with a trigram language model, and add-one smoothing.

Language identification can be considered a ‘solved problem’, with success rates of above 99.8%. However, such systems usually require a minimum of about 50 bytes of text in order to perform well. This data is not available when working on a single word. Instead, Nwesri*et al.*[14] report low success rates using a modification of Cavnar’s method for Arabic foreign words identification. Qu and Grefenstette [13] used Cavnar’s method for the more limited task of language identification of names, to distinguish English, Japanese and Chinese names in Latin script. Training on 88k English names, 83k Japanese names and 11k Chinese names, they achieved accuracies of 92% (Japanese), 87% (Chinese) and 70% (English).

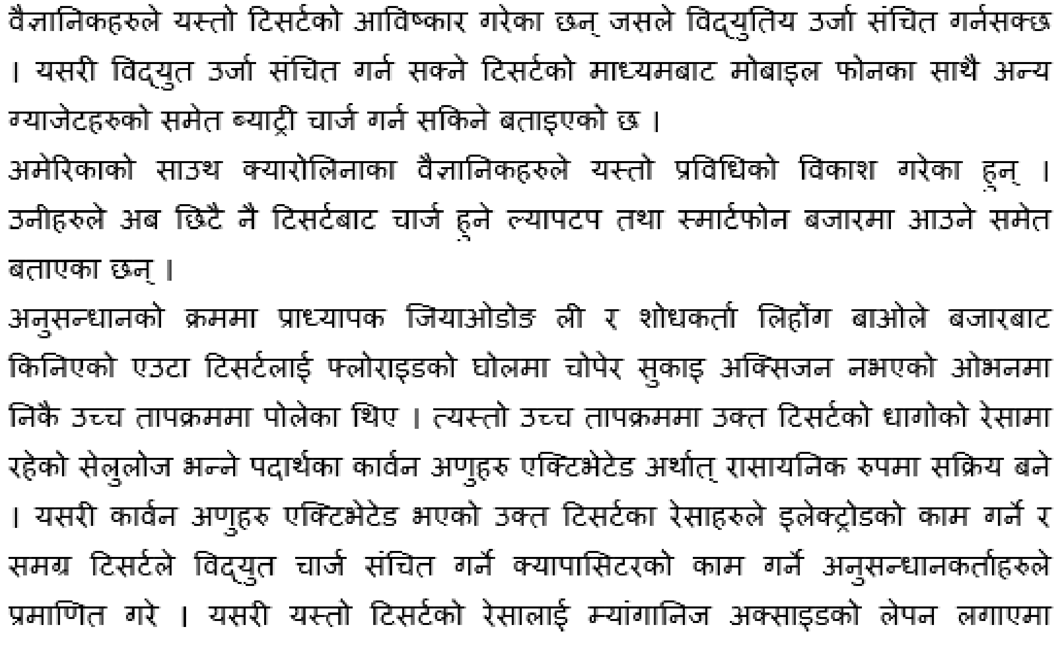
CHAPTER 3

DATA COLLECTION AND IMPLEMENTATION

Here, we concentrate on identifying transliterated and borrowed words in Nepali text. As the most borrowed words come from the English we concentrate on finding borrowed words from such origins.

3.1 Collection of Nepali Corpus

For the purpose of this work we collected the corpus of about 12,000 words (taken from different online news sites). The text in the news site was in the Unicode format which was an advantage as we used Nepali Unicode for this work.



वैज्ञानिकहरुले यस्तो टिसर्टको आविष्कार गरेका छन् जसले विद्युतिय उर्जा संचित गर्नसक्छ । यसरी विद्युत उर्जा संचित गर्न सक्ने टिसर्टको माध्यमबाट मोबाइल फोनका साथै अन्य ग्याजेटहरुको समेत ब्याट्री चार्ज गर्न सकिने बताइएको छ ।

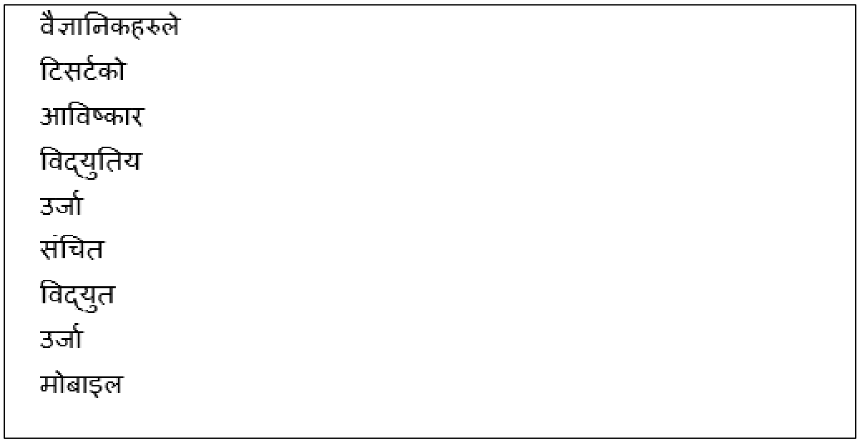
अमेरिकाको साउथ क्यारोलिनाका वैज्ञानिकहरुले यस्तो प्रविधिको विकाश गरेका हुन् । उनीहरुले अब छिटै नै टिसर्टबाट चार्ज हुने ल्यापटप तथा स्मार्टफोन बजारमा आउने समेत बताएका छन् ।

अनुसन्धानको क्रममा प्राध्यापक जियाओडोङ ली र शोधकर्ता लिहोंग बाओले बजारबाट किनिएको एउटा टिसर्टलाई फ्लोराइडको घोलमा चोपेर सुकाइ अक्सिजन नभएको ओभनमा निकै उच्च तापक्रममा पोलेका थिए । त्यस्तो उच्च तापक्रममा उक्त टिसर्टको धागोको रेसामा रहेको सेलुलोज भन्ने पदार्थका कार्वन अणुहरु एक्टिभेटेड अर्थात् रासायनिक रुपमा सक्रिय बने । यसरी कार्वन अणुहरु एक्टिभेटेड भएको उक्त टिसर्टका रेसाहरुले इलेक्ट्रोडको काम गर्ने र समग्र टिसर्टले विद्युत चार्ज संचित गर्ने क्यापासिटरको काम गर्ने अनुसन्धानकर्ताहरुले प्रमाणित गरे । यसरी यस्तो टिसर्टको रेसालाई म्यांगानिज अक्साइडको लेपन लगाएमा

Figure 3.1: Sample corpus

3.2Tokenization

Tokenization is the process of breaking a stream of text up into words, phrases, symbols or other meaningful elements called tokens. The list of token becomes input for further processing. In tokenization, the tokenizer, also called "Lexer" or "Scanner" which takes the raw source text and breaks it into the reserved words, constants, identifier and symbols that are defined in the language. These tokens so found are collected and assigns the possible tags. This assignment may be by simple look up or morphological analysis. The lexicon is usually extracted from pre tagged corpora. This lexicon is referred as dictionary in this dissertation work. This phase prepare the list of word with their possible POS tags.



वैज्ञानिकहरुले
टिसर्टको
आविष्कार
विद्युतिय
उर्जा
संचित
विद्युत
उर्जा
मोबाइल

Figure 3.2: Tokenized Corpus

3.3 Extracting a list of Candidate Affixes

The terms prefix and suffix loosely denote any affixes generally found at the beginning or end of the words. To extract candidate suffixes, each word are scanned from the end of the words and considering every possible suffix in order of increasing length. The following table shows the top 9 suffix list that we have extracted by using string matching algorithm.

S.N.	Suffix
1	को
2	का

3	मा
4	ले
5	लाई
6	बाट
S.N.	**Suffix**
7	त
8	ी
9	ै

Table 3.1 suffix list

Top 9 scoring suffix list is illustrated in Table 4.1

3.3.1 String Matching Algorithm

The string matching algorithm used in this work is based on the simple idea that most transliterated English words have suffixes. In this algorithm we break the transliterated word of length N into two words of length P and Q, where Q is the maximum length of the possible suffix. The sub-word of length Q is again subdivided into words of length ranging from Q to 1. These words are then searched in an array having all possible suffixes of Nepali language one by one. If any word is found in that array the original word of length N is returned otherwise the matched word is deducted from the original word and resulting word is returned.

Algorithm Steps for word युजरलाई

1. Length of the tokenis calculated

   N= Length(युजरलाई)

2. The maximum length(Q) of Nepali suffix is 4 so the token is divided into two words of length N-Q and Q

   Part1= युज and Part2 =रलाई

3. Part2 is again divided into 4 words of length 4,3,2 and 1 respectively

   $\text{Part}_{2,1}$=रलाई

   $\text{Part}_{2, 2}$ =लाई

   $\text{Part}_{2, 3}$ =ाई

   $\text{Part}_{2, 4}$=ई

4. Now each part is searched in an array containing all the Nepali suffix.

   a) If $\text{Part}_{2,i}$ found in the array

Return Original Word- $Part_{2,i}$

Else return Original Word

Here, लाई is found in the array so the Affix युजर is returned.

3.4 Counting Syllables

The foreign words in the Nepali Language are mostly the transliterated English words, so in this work we have applied the English syllabification rule. After the candidate Affix of the token is separated the number of syllables present in a word are found at this stage. For this we have applied the following rule:

1. Count the number of vowels in the word.
2. Subtract any silent vowels in a word.
3. Subtract every vowel from diphthong. (Diphthong only count as single vowel sound.)
4. The number of vowel sound left is the same as the number of syllables.

In word `Compose (कमपोज),` e issilent at the end, so the number of the vowel remaining is 2 hence this word has two syllables as कमandपोज**.**

3.5Splitting rule for Transliterations

I. Divide between two middle consonants.

Split up words that have two middle consonants. For example: hap/pen, bas/ket, let/ter, sup/per, din/ner, and Den/nis. The only exceptions are the consonant digraphs. Never split up consonant digraphs as they really represent only one sound. The exceptions are "th", "sh", "ph", "th", "ch", and "wh".

II. Usually divide before a single middle consonant.

When there is only one syllable, you usually divide in front of it, as in:"o/pen", "i/tem", "e/vil", and "re/port". The only exceptions are those times when the first syllable has an obvious **short sound**, as in "cab/in".

III. Divide before the consonant before an "-le" syllable.

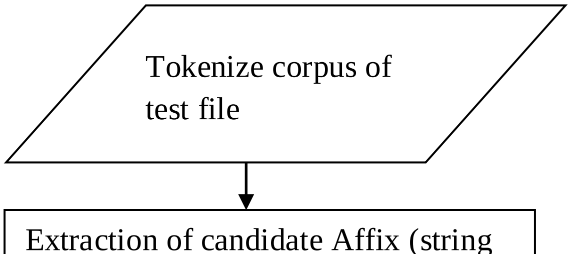
When you have a word that has the old-style spelling in which the "-le" sounds like "-el", divide before the consonant before the "-le". For example: "a/ble", "fum/ble", "rub/ble" "mum/ble" and "thi/stle". The only exceptions to this are "ckle" words like "tick/le".

IV. Divide off any compound words, prefixes, suffixes and roots which have vowel sounds.
Split off the parts of compound words like "sports/car" and "house/boat". Divide off prefixes such as "un/happy", "pre/paid", or "re/write". Also divide off suffixes as in the words "farm/er", "teach/er", "hope/less" and "care/ful". In the word "stop/ping", the suffix is actually "-ping" because this word follows the rule that when you add "-ing" to a word with one syllable, you double the last consonant and add the "-ing".

3.6 Algorithm for tagging syllable structureand classification

1. N be the length of each syllable in the word.
2. If (N==1), if the character in the syllable is vowel, tag the syllable as V otherwise the syllable structure is C. If syllable structure is C, classify the original token as foreign word.
3. If (N==2), then the possible character combination in the syllable is CV, VV, VC and CC. If the syllable structure is CC, classify the original token as foreign word.
4. If (N==3), then the valid Nepali syllable structure are CVV, CCV, and CVC. If the syllable structure is not among these, classify the original token as foreign word.
5. If (N==4), then the valid Nepali syllable structure are CCVV, CCCV, CVC,CCVC. If the syllable structure is not among these, classify the original token as foreign word.
6. If (N==5), then the valid Nepali syllable structure are CCCVC. If the syllable is not in this format, classify the original token as foreign word.
7. If (N>5) classify the original token without finding the syllable structure as the maximum characters in the Nepali syllable is 5.

3.7 Flowchart for foreign word detection



3.8 PHP

In this work we have used PHP to implement the algorithms used for foreign word detection. PHP is an open source server-side scripting language used in Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document rather than calling an external file to process data. The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications.

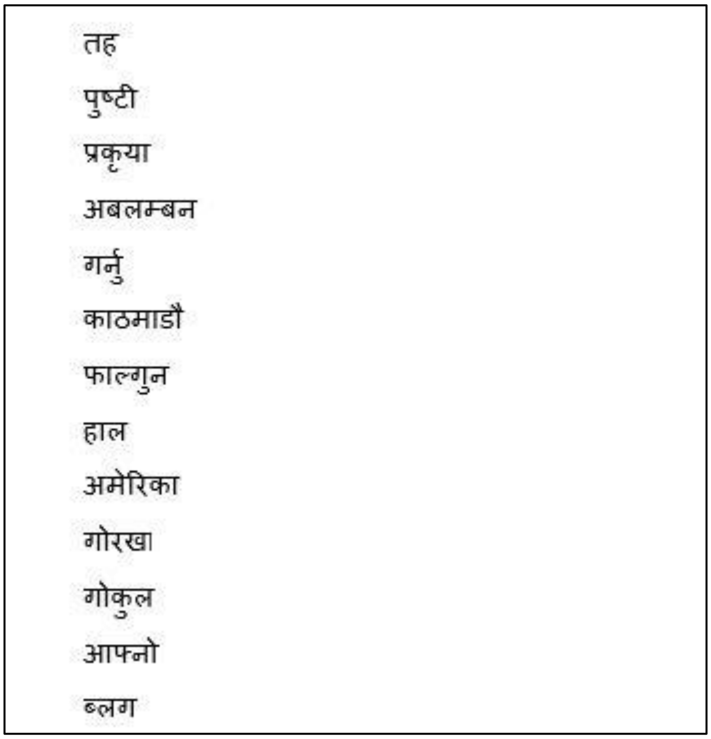
We have used array data structure in PHP and XAMPP. Arrays are sets of data which can be defined in a PHP Script. Arrays can contain other arrays inside of them without any restriction (hence building multidimensional arrays). Arrays can be referred to as tables or hashes. XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

CHAPTER 4

TESTING AND ANALYSIS

4.1 Dictionary

A Nepali Dictionary of about 7000 commonly used words(taken from different online news sites) is created for filtering out the Nepali text from the test file. The resultant words are subject to the rule based classification.

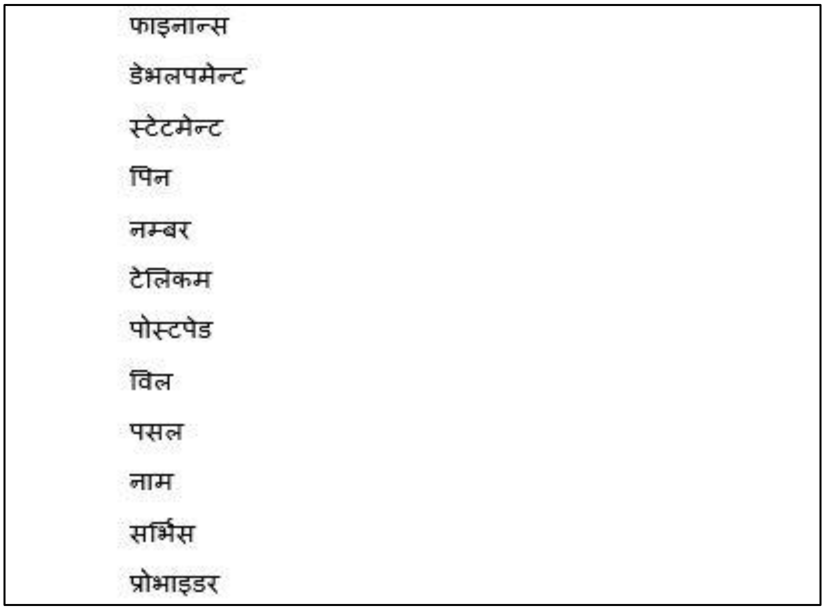


तह
पुष्टी
प्रकृया
अबलम्बन
गर्नु
काठमाडौ
फाल्गुन
हाल
अमेरिका
गोरखा
गोकुल
आफ्नो
ब्लग

Figure 4.1 : Model of Dictionary

4.2 Test Data

Test data are prepared from different news sites like onlinekhabar, mysansar, setopati, hamrakura, ratopati, nepalpati. Text obtained from the category Science and technology, Information technology, Sports News, Feature Articles etc was chosen for testing the algorithm due to the heterogeneous nature of these texts and hence the perceived better representation of the language. In this research work we have done 6 experiments on the basis of size of the test data. Each test data file has more than 5000 words.After removing duplicate words and filtering out the Nepali words from the dictionary the number of unknown words reduced drastically.



फाइनान्स
डेभलपमेन्ट
स्टेटमेन्ट
पिन
नम्बर
टेलिकम
पोस्टपेड
विल
पसल
नाम
सर्भिस
प्रोभाइडर

Fig 4.2: Model of Test data

4.3 Experiment Results

S.N	Word	Syllable	Syllable structure	Remark
1	फोन	फोन	CVC	Nepali syllable (False positive)
2	इपेपर	इ+पे+पर	V, CV, CC	Foreign syllable CC
3	इन्टरनेट	इन+टर+नेट	VC, CC, CVC	Foreign syllable CC
4	डायरी	डायरी	CVVCV	Foreign syllable CVVCV
5	डिजाइन	डि + जाइन	VC, CVVC	Foreign syllable CVVC
6	मोनिटर	मो + नि + टर	CV, VC, CC	Foreign syllable CC
7	फिलिप्स	फि + लि+ प्स	VC, VC, CC	Foreign syllable CC
8	डिजिकम	डि + जि + कम	VC, VC, CC	Foreign syllable CC
9	उफर	उ + फर	V, CC	Foreign syllable CC
10	डिपार्टमेन्ट	डि+ पार्ट + मेन्ट	VC, CVCC, CVCC	Foreign syllable CVCC
11	साइट	साइट	CVVC	Foreign syllable CVVC
12	क्लास	क्लास	CCVC	Nepali syllable (False positive)
13	ब्लक	ब्लक	CCC	Foreign syllable CCC

14	आइकन	आइ + कन	VV, CC	Foreign syllable CC
15	प्रोजेक्ट	प्रो + जेक्ट	CCV, CVCC	Foreign syllable CVCC
16	स्मार्ट	स्मार्ट	CCVCV	Foreign syllable CCVCV
17	लोकेसन	लोके + सन	CVCV, CC	Foreign syllable CVCV and CC
18	एन्ड्रोइड	एन + ड्रोइड	VC, CCVCC	Foreign syllable CCVCC
19	इन्फरमेसन	इन + फर + मे + सन	VC, CC, CV, CC	Foreign syllable CC
20	हाउस	हाउस	CVVC	Foreign syllable CVVC
21	सर्च	सर्च	CCC	Foreign syllable CCC
22	डाटा	डा + टा	CV, CV	Nepali syllable (False positive)
23	इन्ट्री	इन्ट्री	VCCCV	Foreign syllable VCCCV
24	पब्लिक	पब + लिक	CC, VCC	Foreign syllable CC
25	भेरिफिकेसन	भे + रि + फि + के + सन	CV, VC, VC, CV, CC	Foreign syllable CC
26	युजर	यु + जर	VV, CC	Foreign syllable CC
27	कम्यूनिकेशन	कम्यूनि + के + शन	CCVVVC, CV, CC	Foreign syllable CC and CCVVVC

Table 4.1 Experiment Results

Where false positive means the given word should be English but the system predicts it as Nepali.This is due to the rule based syllabification approach. For some exceptional words it has been observed that they have same syllable structure for both Nepali and English. We can improve this result by constructing a dictionary for such exceptional words.

The results of experiment are summarized in the table below. These words are experimented by syllabification and rule based classification.

Test File	**Total Words**	**Total Unknown Words**	**Detected**	**Correct Foreignwords**	**Precision**	**Recall**
File1	5609	152	115	95	82.6%	62.5%
File2	6000	141	107	83	77.5%	58.86%
File3	5500	125	94	77	81.9%	61.6%
File4	6100	131	101	86	85.14%	65.6%
File5	6200	133	95	86	90.52%	64.66%
File6	5900	155	119	97	81.5%	62.58%
Average					83.2%	62.6%

Table 4.2 Result reported in terms of precision (P), and recall (R)

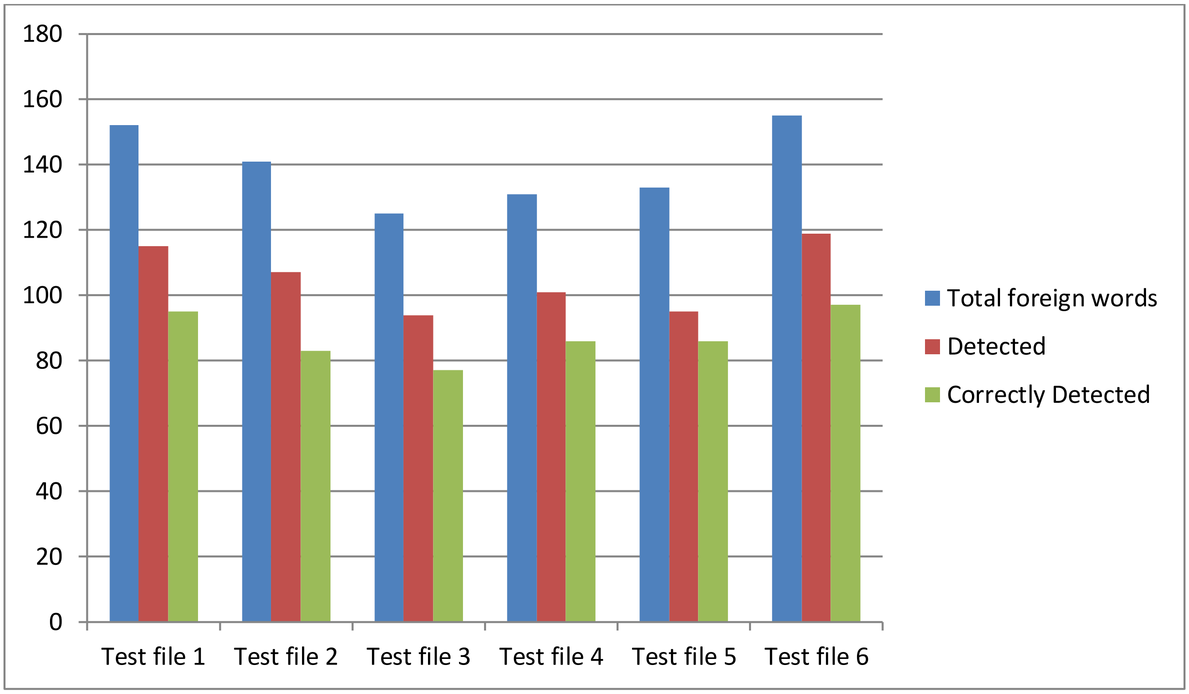


Fig 4.3 Bar chart for Test File

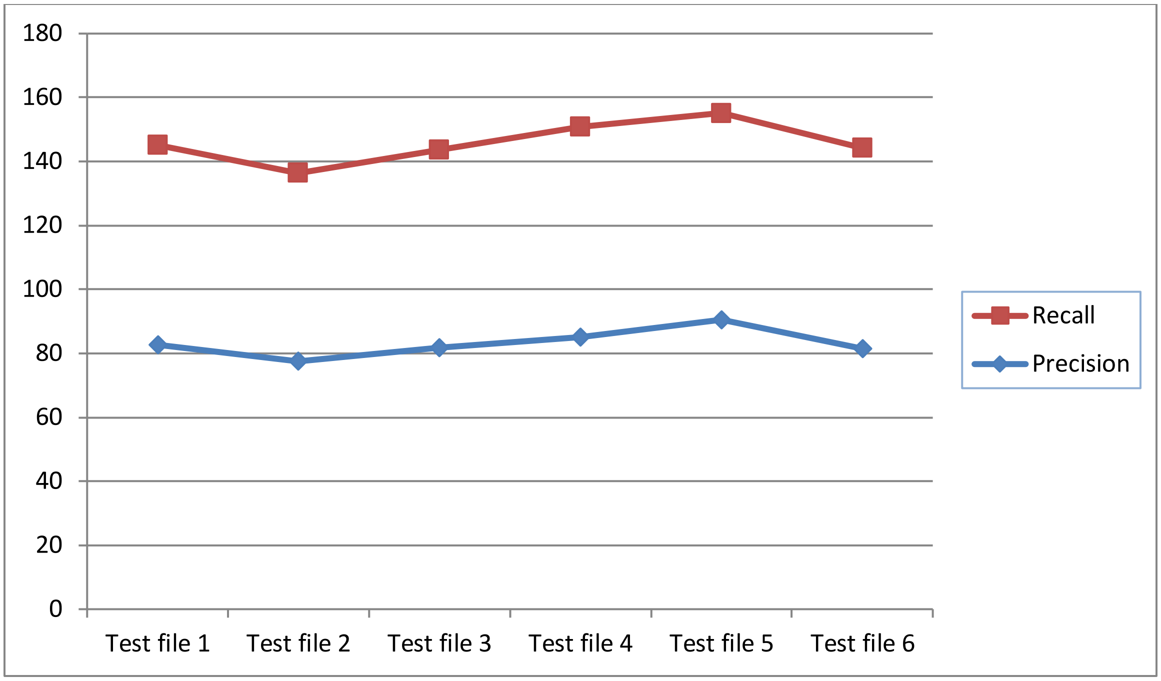


Fig 4.4: Line chart showing Precision and Recall

4.4 Findings

The English words that are transliterated into Nepali words that have vowels at the end have the valid Nepali Syllable structure e.g. फोन, टोन. English syllabification of Transliterated words is useful in classifying the foreign words in Nepali text.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

A method for identification of transliterated words in Nepali text is presented. The preliminary result of experiment shows the satisfactory result. The method is based on syllable tagging approach which assigns tags to individual syllables of words. So it is supervised rule based approach where each syllable structure is matched with the valid Nepali syllable structure. If allthe syllable structure is similar to the valid Nepali syllable structure such word is classified as Nepali otherwise it is classified as foreign word.

Foreign word extraction is very hard task. In addition to statistical information, it requires supporting knowledge of morphological, syntactic, semantic, word type specific and common sense. A word segmented and tagged corpus is essential for the success of the whole research. Since there is no previous work in this field has been done, this work will be an important reference for further research. The result of experiment was 83.2% precision and 62.6% recall.

5.2 Recommendation

In this dissertation, the well-known Nepali words are filtered out by matching the test file with the dictionary.The dictionary used in this work is minimal and the classification was done based only on the rules of syllables, so the result is around 83.2% precision and 62.6% recall. To get accuracy, one can do further research on the other method of classification and increase the size of dictionary. Classification approaches like Neural Network, Support Vector Machine (SVM), Hidden Markov Model (HMM), NaïveBayseian can be used. This research can be further extended with increased size of data used, corpus with part of speech tagging. In future, the research can be done using Nepali syllable structure for word segmentation and without using the dictionary.The findings of the research suggest foreign word extraction may mainly be used for cross language information retrieval, named entity recognition, machine translation, etc.

REFERENCES

[1] Kil-Soon Jeong, Yun-Hyung Kwon, Sung Hyun Myaeng, *Construction of Equivalence Classes of Foreign Words through Automatic Identification and Extraction*, 2004

[2] C.H.Chen and C.C.Hsu, *Synonyms Extraction Using Web Content Focused Crawling,* AIRS 2008, LNCS 4993, pp. 286-297

[3] Byung-Ju Kang and Key-Sun Choi. *Two Approaches for the Resolution of Word Mismatch Problem caused by English Words and Foreign Words in Korean Information Retrieval,* Advanced Information Technology Research Center, Korea Terminology Research Center for Language and Knowledge Engineering, pp.133-140, 2003

[4] Keh-Jiann Chen, Wei-Yun Ma, *Unknown Word Extraction for Chinese Documents,* Institute of Information Science, Academia Sinica, 2001

[5] Byung -Ju Kang, Key -Sun Choi *Effective Foreign Word Extraction for Korean Information Retrieval* Information Processing and Management 38 (2002)pp.91-109, Division of Computer Science, Department of Electrical Engineering & Computer Science, Advanced Information Technology Research Center (AITrc), Korea Terminology Research Center for Language and Knowledge Engineering (KORTERM), Korea Advanced Institute of Science and Technology.

[6] D. Jurafsky,J.H. Martin, *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,* University of Colorado, Boulder,(Pearson Education, 2000) pp 343.

[7] A.Hardie, *The Computational Analysis of Morphosyntactic Categories in Urdu*,(PhD Thesis, Department of Linguistics and Modern English Language, Lancaster University, 2003)

[8] Bal Krishna Bal, MadanPuraskarPustakalaya, Nepal :*Structure of Nepali Grammar working paper (2004-2007) pg no 333-334*

[9] S.P. Srivastav :*Nepali in West Bengal chapter 5 Appendix –I pp.157-237* LSI West Bengal

[10] C.Sharma, J. Talukdar, Prof. P.H. Talukdar :*A Rule Based Algorithm for Automatic Syllabification of a Word of Bodo Language* Vol 1, No.2 Sep-Oct 2012 International Journal of Computing, Communications and Networking

[11] Oh.J, Choi, K.: *A Statistical model For automatic extraction of Korean transliterated foreign words*. International Journal of Computer Proc. Of Oriental Languages 16 (2003)

[12] S. Bartlett and G.Kondrak and C. Cherry :*On the Syllabification of Phonemes* The 2009 Annual Conference of the North America Chapter of the ACL, pages 308-316 (2009)

[13]Y.Goldberg and M.Elhadad :*Identification of Transliterated Foreign Words in Hebrew Script* Computer Science Department Ben Gurion University of the Negev (2008)

[14] Nwesri. A.F., Tahaghoghi, S.Scholer, F.*: Capturing out-of-vocabulary words in Arabic* text.(2006)

BIBLIOGRAPHY

BhusanChhetri , Krishna BikramShah *Nepali Text to Speech System Synthesis System using ESONOLA Method of Concatenation* International Journal of Computer Applications Vol 62-No.2, January 2013

Hafiz Musa, RabiahA.Kadir, AzreenAzman, M.Taufik Abdullah *Syllabifacation Algorithm based on Syllable Rules Matching for Malay Language,* Recent Research in Applied Computer and Applied Computational Science, Department of Multimedia, Faculty of Computer Science and Information Technology Universiti Putra Malaysia

Jin-Shea Kuo, Ying-kuei Yang *Incorporating Pronunciation Variation into Extraction of Transliterated-term Pairs from Web Corpora* Journal of Chinese Language and Computing 15 (1):(33-44)

Shankar Ananthakrishnan*Statistical Syllabification of English Phoneme Sequences using Supervised and Unsupervised Algorithms*ATerm Project Dec, 2004

SrishteeGurung and IshworThapa *Text to Speech Generation in Nepali* pg.91-134 14th March 2008

Susan Bartlett and GrzegorzKondrak and Colin Cherry *Automatic Syllabification with structured SVMs for Letter-To-Phoneme Conversion* Proceedings of ACL-08:HLT, pages 568-576, Columbus, Ohio, USA, June 2008, Association for Computational Linguistics

T.A.Hall*English Syllabification as the Interaction of Markedness Constraints*StudiaLinguistica 60(1) 2006, pp.1-33 The Author 2006, Journal Compilation

APPENDIX

Code For Implementation

```
<?
$my_file = 'file.txt';
$handle = fopen($my_file, 'w') or die('Cannot open file:  '.$my_file);
$in_file = 'syllablewords.txt';
$handle1 = fopen($in_file, 'r') or die('Cannot open the file:  '.$my_file);
$vowels=array("आ","ऐ","ए","इ","ई","ओ","उ","ऊ","ऋ","ॐ","अ","अः","ौ","े","ै","ृ","ु","ू","ि",
"ी","ो","ा","ँ");
$const=array("श","स","द","ध","ग","घ","ह","ज","झ","क","ख","ल","ष","ड","ढ","च","छ","व","ब","भ",
"न","ण","म","ङ","ट","ठ","र","त","थ","य","प","फ","क्ष","त्र","ञ","ज्ञ");
$sylstr=array("V","VV","VC","CV","CVV","CCV","CCVV","CCCV","CVC","CCVC","CCCVC");
$filearray=array();
while(! feof($handle1))
 {
 $line=fgets($handle1);
        $trimmed=trim($line);
 $txtword=mb_str_split("$trimmed");
        $trimme=RemoveAdverd($txtword);
        $word=mb_str_split("$trimme");
        print("---------------Processing of word: $trimme---------------");
        echo "</br>";
        BreakSyllable($word,$vowels,$const,$handle);
                echo "</br>";
        echo "</br>";
        //Length($word,$vowels,$const,$handle);
 }
fclose($handle1);
echo"</br>";
functionRemoveAdverd($line)
{
        $adverd=array("लाई","मा","का","बाट","देखि","लागि","को","ले","हरू");
        $flag=0;
        $checkarray=array();
        $cou=count($line);
        $pos=$cou-4;
        $rema=array_slice($line,$pos);
        $rem3=array_slice($rema,1);
        $rem2=array_slice($rema,2);
```

```
$rem1=array_slice($rema,3);
$im1=implode("",$rema);
array_push($checkarray,$im1);
$im2=implode("",$rem3);
array_push($checkarray,$im2);
$im3=implode("",$rem2);
array_push($checkarray,$im3);
$im4=implode("",$rem1);
array_push($checkarray,$im4);
for($i=0;$i<=3;$i++)
{
		if(in_array($checkarray[$i],$adverd))
		{
				$flag=$flag+1;
				if($checkarray[$i]==$im1)
				{
						$cou1=4;
						$pos1=$cou-$cou1;
				$sl1=array_slice($line,0,$pos1);
				$final=implode("",$sl1);
				return $final;
				}
				else if($checkarray[$i]==$im2)
				{
						$cou1=3;
						$pos1=$cou-$cou1;
				$sl1=array_slice($line,0,$pos1);
				$final=implode("",$sl1);
				return $final;
				}
				else if($checkarray[$i]==$im3)
				{
						$cou1=2;
						$pos1=$cou-$cou1;
				$sl1=array_slice($line,0,$pos1);
				$final=implode("",$sl1);
				return $final;
				}
				else if($checkarray[$i]==$im4)
				{
						$cou1=1;
						$pos1=$cou-$cou1;
```

```
            $sl1=array_slice($line,0,$pos1);
            $final=implode("",$sl1);
            return $final;
            }

        }
    }
    if ($flag==0)
            {
                $cou1=0;
                $pos1=$cou-$cou1;
            $sl1=array_slice($line,0,$pos1);
            $final=implode("",$sl1);
            return $final;
            }
}
functionBreakSyllable($string1,$vowels,$const,$handle)
{
    $remove=array("्");
    $string2=array_diff($string1,$remove);
    $string=array_values($string2);
    $couv=0;
    $pos=array();
    print_r($string);
    echo "</br>";
    $countarra=count($string);
    for($i=0;$i<$countarra;$i++)
    {
        if(in_array($string[$i],$vowels))
        {
        $couv++;
        array_push($pos,$i);
        }

    }
    if($couv==1)
    {
        if(($string[$countarra-1]=="र")  || ($string[$countarra-1]=="ल"))
        {
                                        $part1=array_slice($string,0,2);
                                        print_r($part1);
                                            echo "</br>";
```

```
                            $part2=array_slice($string,2);

Length($part1,$vowels,$const,$handle);

Length($part2,$vowels,$const,$handle);
    }


    else if($pos[0]>0)
    {
    $aa=$pos[0];
    if(($string[$aa]=="ि"）   || ($string[$aa]=="ी"））
    {
        if(($countarra-$aa)==1)
    {
                            $part1=array_slice($string,0,2);
                            print_r($part1);
                                echo "</br>";
                            $part2=array_slice($string,2);
                            print_r($part1);
                                echo "</br>";

Length($part1,$vowels,$const,$handle);

Length($part2,$vowels,$const,$handle);
    }
    else
    {
                            $part1=array_slice($string,0,1);
                            print_r($part1);
                                echo "</br>";
                            $part2=array_slice($string,1,2);
                            print_r($part2);
                                echo "</br>";
                            $part3=array_slice($string,$aa+1);
                            print_r($part3);
                                echo "</br>";

Length($part1,$vowels,$const,$handle);

Length($part2,$vowels,$const,$handle);
Length($part3,$vowels,$const,$handle);
    }

    }
    else
    {
                            $part1=array_slice($string,0,2);
                            print_r($part1);
                                echo "</br>";
                            $part2=array_slice($string,2);
                            print_r($part1);
                                echo "</br>";

Length($part1,$vowels,$const,$handle);

Length($part2,$vowels,$const,$handle);

    }
    }
    else
    {
    print("it has only one syllable:");
    echo "</br>";
    Length($string,$vowels,$const,$handle);
    }
}
if($couv==2)
{
        if(($pos[1]-$pos[0])==1)
        {
            if($pos[0]==0)
            {
                $part1=array_slice($string,0,2);
                    Length($part1,$vowels,$const,$handle);
                $part2=array_slice($string,2);
                    Length($part2,$vowels,$const,$handle);
            }
            else
            {

            print("it has one syllable:");
    Length($string,$vowels,$const,$handle);
                {
            }
            else
            {
                $p1=$pos[0];
                $p2=$pos[1];
                $part1=array_slice($string,0,$p1+1);
                $part2=array_slice($string,$p2-1);
                Length($part1,$vowels,$const,$handle);
                Length($part2,$vowels,$const,$handle);
//              $prt1=implode("",$part1);
//              $prt2=implode("",$part2);
//              print($prt1);
//              echo "</br>";
//              print ($prt2);
            }

    }
    if($couv==3)
    {
            print("it has three syllable:");
                $p1=$pos[0];
                $p2=$pos[1];
                $p3=$pos[2];
                if($p2-$p1=0)
                {
                    $part1=array_slice($string,0,2);
                    $part2=array_slice($string,2);
                Length($part1,$vowels,$const,$handle);
                Length($part2,$vowels,$const,$handle);
                }
                else if($p3-$p2=0)
                {
                    $part1=array_slice($string,0,$p1+1);
                    $part2=array_slice($string,$p2-1);
                Length($part1,$vowels,$const,$handle);
                Length($part2,$vowels,$const,$handle);
                }

                else{
                    $part1=array_slice($string,0,$p+1);
                    $rem=array_slice($string,$p1+1);
                    $part2=array_slice($rem,0,$p2+1);
                    $part3=array_slice($string,$p2+1);
                Length($part1,$vowels,$const,$handle);
                Length($part2,$vowels,$const,$handle);
                Length($part3,$vowels,$const,$handle);
                }

    }
    else
    {
        print "This is foreign word";
    }

}
functionmb_str_split( $string )
    {
   # Split at all position not after the start: ^
   # and not before the end: $
returnpreg_split('/(?<!^)(?!$)/u', $string );
    }
function Length($word,$vowels,$const,$handle)
{
}
functionChooseFunction($word,$vowels,$const,$handle)
{
            $num=count($word);
                if($num==1)
                {
                SyllableSplitOne($word,$vowels,$const,$handle);
                }
                else if($num==2)
                {
                SyllableSplitTwo($word,$vowels,$const,$handle);
                }
                else if($num==3)
                {
                SyllableSplitThree($word,$vowels,$const,$handle);
                }
                else if($num==4)
                {
                SyllableSplitFour($word,$vowels,$const,$handle);
                }
                else if($num==5)
                {
                    SyllableSplitFive($word,$vowels,$const,$handle);
                }
                else
                {
                    echo "rejected";
                }


}
functionSyllableSplitOne($word,$vowels,$const,$handle)
{
    $v=array();
    /*V */
    $i=0;
                if((in_array($word[$i],$vowels)))
            {
                        array_push($v,$word[$i]);

            }
            echo "-----------------------------------------------------------------------------------";
            echo "</br>";
            if(count($v)>0)
            {
            print("The V syllable is:");
            print_r($v);
            echo "</br>";
            }
    }
functionSyllableSplitTwo($word,$vowels,$const)
{
    $v=array();$vv=array();$vc=array();$cv=array();$c=array();
    /*V VV VC */
    $i=0;
                if(in_array($word[$i],$vowels))
            {
                array_push($v,$word[$i]);
                if(in_array($word[$i+1],$vowels))
                {
                                array_push($v,$word[$i]);
                }
                else
                {
                                array_push($vc,$word[$i]);
                                array_push($vc,$word[$i+1]);
                }
            }
            /*CC CV */
            else if(in_array($word[$i],$const))
            {
                array_push($c,$word[$i]);
                if(in_array($word[$i+1],$const))
                {
                array_push($c,$word[$i+1]);
                $ccc++;
                print("this is a foreign word");
                echo "</br>";
                print_r($c);
                echo "</br>";
                }
                else
                {
                        array_push($cv,$word[$i]);
                        array_push($cv,$word[$i+1]);
            }
            }
            echo "----------------------------------------------------------------------------------";
            echo "</br>";
            if(count($vv)>0)
            {
            print("The VV syllable is:");
            print_r($vv);
            echo "</br>";
            }
            if(count($vc)>0)
            {
            print("The VC syllable is:");
            print_r($vc);
            echo "</br>";
            }
            if(count($cv)>0)
            {
```

```
                        print("The CV syllable is:");
                        print_r($cv);
                        echo "</br>";
                        }
    }

    functionSyllableSplitThree($word,$vowels,$const)
{
    $v=array();$vv=array();$vc=array();$cv=array();$cvv=array();$ccv=array();$cvc=array();$c=array()
;
    /*V VV VC */
    $i=0;
                            if(in_array($word[$i],$vowels))
                        {
                            array_push($v,$word[$i]);
                            if(in_array($word[$i+1],$vowels))
                            {
                                                            array_push($v,$word[$i]);
                            }
                            else
                            {
                                                            array_push($vc,$word[$i]);
                                                            array_push($vc,$word[$i+1]);
                            }
                        }
                        /*CC CV */
                        else if(in_array($word[$i],$const))
                        {
                            array_push($c,$word[$i]);
                            if(in_array($word[$i+1],$const))
                            {
                            array_push($c,$word[$i+1]);
                                                            if(in_array($word[$i+2],$const))
                                                            {

    array_push($c,$word[$i+2]);

                                                            }
                                                            else
                                                            {

    array_push($ccv,$word[$i]);
array_push($ccv,$word[$i+1]);
array_push($ccv,$word[$i+2]);
                            }
            }
            else
            {
                    array_push($cv,$word[$i]);
                    array_push($cv,$word[$i+1]);
                    if(in_array($word[$i+2],$const))
                            {

                    array_push($cvc,$word[$i]);
                    array_push($cvc,$word[$i+1]);
                    array_push($cvc,$word[$i+2]);
                            }
                            else
                            {

                    array_push($cvv,$word[$i]);
                    array_push($cvv,$word[$i+1]);
                    array_push($cvv,$word[$i+2]);
                            }

        }
        }
        echo "-----------------------------------------------------------------------------------";
        echo "</br>";
        if(count($cvv)>0)
        {
        print("The CVV syllable is:");
        print_r($cvv);
        echo "</br>";
        }
        if(count($cvc)>0)
        {
        print("The CVC syllable is:");
        print_r($cvc);
        echo "</br>";
        }
        if(count($ccv)>0)
        {
        print("The CCV syllable is:");
```

```
            print_r($ccv);
            echo "</br>";
            }
    }
functionSyllableSplitFour($word,$vowels,$const,$handle)
{
    $v=array();$vv=array();$vc=array();$cv=array();$cvv=array();$ccv=array();$ccvv=array();$cccv=ar
ray();
    $cvc=array();$ccvc=array();$c=array();
    /*V VV VC */
    $i=0;
                if(in_array($word[$i],$vowels))
            {
                array_push($v,$word[$i]);
                if(in_array($word[$i+1],$vowels))
                {
                                array_push($vv,$word[$i]);
                                array_push($vv,$word[$i+1]);
                }
                else
                {
                                array_push($vc,$word[$i]);
                                array_push($vc,$word[$i+1]);
                }
            }
            /*CC CV */
            else if(in_array($word[$i],$const))
            {
                array_push($c,$word[$i]);
                if(in_array($word[$i+1],$const))
                {
                array_push($c,$word[$i+1]);
                                if(in_array($word[$i+2],$const))
                                {

    array_push($c,$word[$i+2]);

    if(in_array($word[$i+3],$vowels))
                {
    array_push($cccv,$word[$i]);
    array_push($cccv,$word[$i+1]);
    array_push($cccv,$word[$i+2]);
array_push($cccv,$word[$i+3]);
if(in_array($word[$i+4],$const))
{
array_push($cccvc,$word[$i]);
array_push($cccvc,$word[$i+1]);
array_push($cccvc,$word[$i+2]);
array_push($cccvc,$word[$i+3]);
array_push($cccvc,$word[$i+4]);
}
else
{
        echo "This is foreign word";
        echo "</br>";
}
}
                                                        }
                                                        else
                                                        {

array_push($ccv,$word[$i]);
array_push($ccv,$word[$i+1]);
array_push($ccv,$word[$i+2]);
if(in_array($word[$i+3],$vowels))
{
array_push($ccvv,$word[$i]);
array_push($ccvv,$word[$i+1]);
array_push($ccvv,$word[$i+2]);
array_push($ccvv,$word[$i+3]);
}
else
{
array_push($ccvc,$word[$i]);
array_push($ccvc,$word[$i+1]);
array_push($ccvc,$word[$i+2]);
array_push($ccvc,$word[$i+3]);
}
                                                        }
                        }
                        else
                        {
                                        array_push($cv,$word[$i]);
                                        array_push($cv,$word[$i+1]);
                        if(in_array($word[$i+2],$const))
                                {
    array_push($cvc,$word[$i]);
    array_push($cvc,$word[$i+1]);
    array_push($cvc,$word[$i+2]);
                                }
                                else
                                {
    array_push($cvv,$word[$i]);
    array_push($cvv,$word[$i+1]);
    array_push($cvv,$word[$i+2]);
                                }
            }
            }
            $wr=implode($word);
            //fwrite($handle, $wr);
            //fwrite($handle,"\n");
            echo "----------------------------------------------------------------------------------------";
            echo "</br>";
            if(count($cccv)>0)
            {
            print("The CCCV syllable is:");
            print_r($cccv);
            echo "</br>";
            }
            if(count($ccvv)>0)
            {
            print("The CCVV syllable is:");
            print_r($ccvv);
            echo "</br>";
            }
            if(count($ccvc)>0)
            {
            print("The CCVC syllable is:");
            print_r($ccvc);
            echo "</br>";
            }
    }

functionSyllableSplitFive($word,$vowels,$const,$handle)
{
    $cccvc=array();
```

```
        $c=array();
        /*V VV VC */
        $i=0;
                    if((in_array($word[$i],$const)) && (in_array($word[$i+1],$const)) &&
(in_array($word[$i+2],$const)) && (in_array($word[$i+3],$vowels)) && (in_array($word[$i+4],$const)))
                {

        array_push($cccvc,$word[$i]);
        array_push($cccvc,$word[$i+1]);
        array_push($cccvc,$word[$i+2]);
        array_push($cccvc,$word[$i+3]);
        array_push($cccvc,$word[$i+4]);
                }

                echo "--------------------------------------------------------------------------------------------------";
                echo "</br>";
                if(count($cccvc)>0)
                {
                print("The CCCVC syllable is:");
                print_r($cccvc);
                echo "</br>";
                }
        }
        print "$ccc";
?>
```