N-GRAM BASED ALGORITHM FOR DISTINGUISHING BETWEEN HINDI AND SANSKRIT TEXTS

Project report submitted in complete fulfillment of the requirement for the degree of

BACHELORIN TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

By

MANOJ KUMAR (131253)

RADHA AGNIHOTRI (131334)

Under the supervision of

Dr. RAJNI MOHANA

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT.

May, 2017

CERTIFICATE

I hereby declare that the work presented in this report entitled " **N-Gram Based Algorithm ForDistinguishing Between Hindi And Sanskrit Texts** "in partial fulfilment of the requirementsfor the award of the degree of **Bachelor of Technology** in **Computer Science andEngineering/Information Technology** submitted in the department of Computer Science &Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2016 to December 2016 under the supervision of **Dr. Rajni Mohana**, Assistant Professor in the department of Computer Science And Information Technology. The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Manoj Kumar (131253)

Radha Agnihotri (131334)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Rajni Mohana

Assistant Professor

Department of Computer Science and Information Technology

Dated:

ACKNOWLEDGEMENT

Apart from the efforts, the success of any project depends largely on the encouragement and guidelines of many others. Therefore, we take the opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We would also like to show our appreciation to our project guide **Dr. Rajni Mohana**. Without her able guidance, tremendous support and continuous motivation the project work would not be carried out satisfactory. Her kind behaviour and motivation provided us the required courage to complete our project

We would also like to thank our Director, Dean and Head of department of computer science for their continuous support and guidance. Special thanks to our project panel because it was their regular concern and appreciation that made this project carried out easily and satisfactorily.

Manoj Kumar

Radha Agnihotri

Date: -

Table of Content

TOPICS **Page No.**

LIST OF ABBREVIATIONS

ABBREVIATION	**FULL FORM**
OCR	Optical Character Recognition
LI	Language Identification
TDIL	Technology development for Indian languages
DEIT	Department of Electronics and Information technology
NLTK	Natural language tool kit.
LM	Language Model
FAQ	Frequently ask question
ISCII	India standard code for Information interchange
MB	Mega byte
OOPS	Object oriented programming

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Language identification (LI) is an essential and integral part of “natural language processing”. Several machine learning approaches have been proposed so far for addressing this sort of a problem. “Language Identification “can be defined as the process of automatically determining the language(s) in which the content has been written in any document (web page, text document). Due to the rampant use of internet, identification of language has become a necessary pre-processing step for a variety of applications such as machine translation, linguistic corpus creation, Part-of-Speech tagging, accessibility of social media or user-generated content, search engines, supporting low-density languages and information extraction in addition to processing multilingual documents. In a multilingual country like India,“Language Identification” has wider scope to bridge the digital rift between different language users. This project presents a brief overview of the challenges involved in the automatic identification of language as well as existing methodologies and some of the tools available identification. The process of “Text categorization” is a fundamental task in document processing that allows the automated handling of large streams of documents in the electronic form. It must work in a reliable manner on all inputs, and therefore must tolerate problems of auto-identification up to some extent. Here, we describe an “N-gram-based approach” to text categorization that is capable of distinguishing between Hindi and Sanskrit words. The system is small, speedy and robust. It has worked well for language classification, achieving an accuracy of 94.8%.

Chapter-1 INTRODUCTION

1.1 INTRODUCTION

In recent times, a lot of research has been carried out in the fields of multilingual textual data and data processing. This is due to many reasons- the development of communication infrastructure and the Internet, an increasing collection of networked and universally distributed data, the increasing no. of people that are connected to the global network and whose mother tongue is not English. This has created a need to organize and process a huge volume of data as it is very costly in terms of time and personnel employed. There are inflexible methods, so we try to develop automatic methods to auto identify the textual language. Electronic documents come from a wide variety of sources where most are generated with various word processing software packages, and are subjected to various kinds of automatic scrutiny, e.g., spelling checkers, as well as to manual editing and revision. Many other documents, however, do not get the benefit of this kind of scrutiny, and thus may contain significant number of errors. Email messages and bulletin board postings, for example, are often composed on the fly and sent without any inspection and correction. The paper documents that generally are digitally scanned, passed and run through an OCR system will certainly contain some level of recognition errors. It is because of these kinds of documents where further manual inspection and correction is difficult and costly, that there would be the greatest benefit in automatic processing. Text categorization is one of the fundamental kinds of document processing, in which an incoming document is assigned to some pre-existing category.

One of the applications of such a system is to route news articles from a newswire. Another application would be sorting through digitized paper archives. These applications have the following characteristics:

- The categorization must be reliable in spite of textual errors.

- The categorization must be efficient, consuming as little storage and processing time as possible.
- When a given, document does not match any category, or when it falls under two categories, then the categorization must be able to recognize it.

The multilingual sentiment analysis on social media is a good example. The research paper presented by Tromp in the year 2011 shows an extensive experimental analysis about the multilingual sentiment classification that can be performed more accurately when the process is split into four steps. This becomes possible because at each step after LI, models that utilize language specific knowledge can be applied. Certainly, if the language of some text is determined incorrectly then this error will affect the forthcoming steps of the multilingual sentiment analysis and thus compromise the use of the four step procedure. Therefore, it is highly desirable to minimize the error of LI. Consider another example of machine translation where the source text language is not known. The translation cannot even commence without first identifying the source language. Numerous supervised machine learning techniques have been proposed for LI. The most widely accepted approach is to use N-grams given by Cavnar & Trenkle, in 1994. It was shown to be almost 99% accurate for long texts. The experimental results showed 99% accuracy on the collection of various documents written in fourteen different languages. The results suggested that high accuracies can be achieve for texts having a text length of at least four hundred characters but when the language identification is done for documents having less than three hundred characters, the accuracies start to decrease much faster (though still pertaining the level of 93%) with respect to relatively longer texts having at least four hundred characters. Over recent years, with the popularity of social media, including Twitter and social networks, and consequently social media data analysis like opinion mining, the need for accurate LI (but now on short and grammatically-ill text messages) has become well motivated again.

1.2 PROBLEM STATEMENT

There is an increasing need to deal with multi-lingual documents in today's time. In a multilingual society like India, where there are twenty- two official languages, "language identification" has wider scope, and would be a vital step in bridging the digital rift between different language users. Languages in India look similar because they use almost same alphabets and scripts. Language Identification of web pages is quite a challenging task. If multi-lingual documents can be segmented and organized language-wise, it would be very useful. Language identification is particularly useful in the field of information extraction in order to retrieve language specific information and in any library for categorizing materials. Libraries often have to categorize materials whose languages are not known, and hence they rely on tables of frequently occurring words and distinctive letters or characters for identifying languages. But this method will not work if one has to distinguish a language from another with similar orthography. Hence, we present an "N-gram" based method for efficient detection and identification of Sanskrit and Hindi text data which share almost similar scripts. Character based n-grams have been applied to language identification together with, language modelling; frequency profile matching and compression. N-grams have previously been considered in foreign name identification but without rigorous experiments and using ad-hoc techniques. Hammarstrom models "word emission probabilities" with relative frequencies of words. An unsupervised affix detection component models' unseen words and languages which are minimized in a sequence of words. The method is tested with different languages but is not compared to any other method. In general, a string of length "*k"*, padded with blanks, will have "*k*+1" bi-grams," k+1" tri-grams," k+1" quad-grams, and so on. N-gram based matching has had some success in dealing with noisy ASCII input, such as in text retrieval and in a wide variety of other natural language processing applications. The key benefit that N-gram-based matching provides comes from its very nature: since every string is split into small parts, any errors that are present tend to affect only a portion of these parts, leaving the remainder intact. One could get a measure of similarity for the n-grams that are common to two strings which are resistant to a wide variety of textual errors. According to a law the *n*th most common word in a human language text occurs with a frequency that is inversely proportional to *n*. The implication of this law is that there is always a set of words which dominates most of the other words of the language in terms of frequency of use. This is true both of words in general, and of words that are specific to a particular subject. Furthermore, there is a smooth continuum of dominance from most frequent to least. The smooth nature of frequency curves helps us in some ways, because it implies that we do not have to worry too much about specific frequency thresholds.

1.3 OBJECTIVES

There are 22 official languages having twelve scripts, that are being spoken by the Indians because of multilingual regions. In view of the fact that only a small number of people know English in India, others are deprived of the benefits of IT development. The benefits of information technology are better accessible when the software tools and interface systems are available in one’s own language.

Technology Development for Indian Languages (TDIL) Programme initiated by the Department of Electronics and Information Technology (DeitY), Ministry of communications and Information Technology, Government of India has the objective to create information-processing tools and technologies to enhance human machine interaction in Indian languages and also to create and access multilingual knowledge resources.

Our project objective is to understand Natural Language Processing techniques in order to handle multi-lingual text using N-Grams. We have to prepare a corpus and then we’ll be comparing precision and accuracy with standard results.

1.4 METHODOLOGY

In this work, N-grams over Hindi and Sanskrit text data were analysed. The system is trained with different N-gram sets of both the languages. From the training set profiles generated, the system classifies the unknown language given in the testing phase by calculating the similarity measure. Based on the evaluation of different text inputs, the apt N-gram profile which is reliable for distinguishing the two languages was found. For this, initially, a separate corpus of approximately 1 MB size was created for both Hindi and Sanskrit language. Then the language corpus was filtered and sent as parameter to the training profile generator to generate character based as well as word based unigram, bigram and trigram training set for both Hindi and Sanskrit text data. Thus, a separate language profile for both languages was formulated based on N-gram frequency count from the corpus.

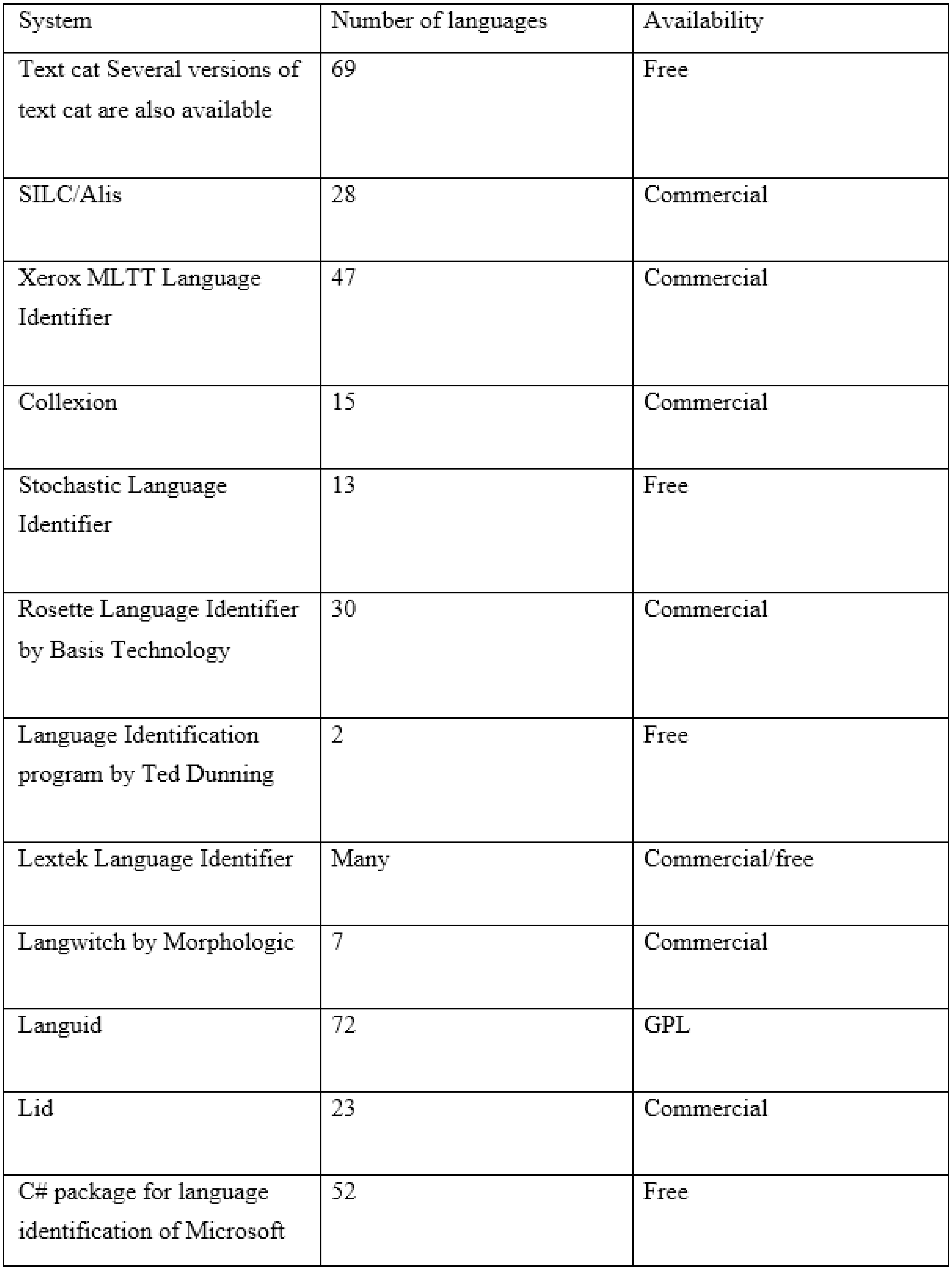
Natural language Tool Kit (NLTK) in Python language [6] was used for this experimentation and result formulation. Once the training set was created, the proposed system was used on random test data for classification and identification of unknown content in the digital online text. This test data was taken from random documents and texts from Internet with sentences in either Hindi or Sanskrit and the results were noted. For that the test data was filtered and sent as parameter to the testing profile generator code to generate character based as well as word based unigram, bigram and trigram training set text data. Then the similarity measure was noted down. The methodology used is further explained in chapter-3. Formula used to calculate the accuracy-:

Accuracy = (*numberofwordsmatched/total no. of words in corpus)* * 100.

TOOLS FOR LI

Research in LI has resulted in the availability of a number of tools for the identification of language(s) automatically [16]. Table 1 lists a few tools for LI available commercially as well as freely.

Table 1.1: Tools for LI



System	Number of languages	Availability
Text cat Several versions of text cat are also available	69	Free
SILC/Alis	28	Commercial
Xerox MLTT Language Identifier	47	Commercial
Collexion	15	Commercial
Stochastic Language Identifier	13	Free
Rosette Language Identifier by Basis Technology	30	Commercial
Language Identification program by Ted Dunning	2	Free
Lextek Language Identifier	Many	Commercial/free
Langwitch by Morphologic	7	Commercial
Languid	72	GPL
Lid	23	Commercial
C# package for language identification of Microsoft	52	Free

1.5 ORGANISATION

Chapter-1 includes the introduction to the project briefly. We have
Introduced the project so as to give the basic idea of what we are doing in the project.

Chapter-2 includes the literature survey. In this section, we have mentioned about the different papers surveyed. We have studied many international journals and conference papers on artificial intelligence before carrying out our project.

Chapter-3 includes the system development. In this section, we have briefly explained our project system model, design, development and formula used.

Chapter-4 includes the analysis. In this section, we are describing the analysis of project model and the computed accuracy.

Chapter-5 includes the conclusion. In this section, we have mentioned about the outcomes of project and the future scope of the work.

Chapter-2 LITERATURE SURVEY

Major portion of research in language identification is parsing the document image to identify the basic script. Mallamma has given a method that identifies and separates text lines of English, Telugu documents and Kannada contained a trilingual document. Also, Padma and Vijaya have done identification of language and script using technique called OCR. Mallikarjun et al. has given a word level script identification that uses global and local features. Deepamala et al. has identified a method based on N-gram for identification of Kannada language and it also describes how to improve the performance using the last word only, instead of using the complete sentence. B. Ahmed et al. has given a logarithmic version of the 'Cumulative Frequency Addition- CFA Bayesian' which is based on N-gram and text-classification algorithm. Because the approach is basically based on the analysing the unigram statistical approach of individual text lines, therefore it needs the segmentation of character or word. Grigory Grefenstette has presented two techniques to identify non-linguistic corpus-derived attributes, trigrams or short words. Tommi et al. has laid emphasis on the language identification task of short text segments using N-gram models.

Sanskrit and Hindi are written using the Devanagari script. Therefore, only script identification is not sufficient. It is mandatory to identify the language irrespective of the script being used. Making it useful for automatic language detection, identification and the classification of documents.

Language identification of text is an initial step process for NLP. Many researchers have been already worked on this direction. A system for an auto identification of text is already proposed by S. Sreejith, Indu.M, Dr Reghu Raj P C [1]. In which a model n-gram techniques from Natural language tool kit is used. The system achieved 99% accuracy. The system was capable of identifying the language of a unknown text taken from online. But it was working for only two Devanagari language Hindi and Sanskrit.

Monica T Makwana and Deepak C Vegda [2] worked on semantic analysis of Indian languages. Study the structure of a sentence is known as syntax. Semantic analysis deals with the grammatical aspects of a structure of a sentence. In this work, a huge amount of corpus was needed to do the semantic analysis. Accuracy of this model was completely based on corpus size.

Andrija Tomovic and Pradrig Janice worked on n-gram based language identification for 20 European languages based on its similarities and dissimilarities of a language code have been assigned to each language. A model attained 100% accuracy with their corpus of integral document. Many work has been done on European language but when it comes to Indian languages the work becomes more tedious.

There are many methods for language identification of long text samples, but the identification of very short strings still is a challenge.
In this paper, there is a test sample of 5-21 characters has been taken for language identification. The Author has compared two methods that are well suited for this task:

a naive Bayes classifier based on character n-gram models

the ranking method by Cavnar and Trenkle (1994).

For the n-gram models, he tested many standard techniques that included current state of the art, and the modified Kneser-Ney interpolation. Conducted Experiments with 281. The identification accuracy improved due to the advanced language model smoothing techniques. Higher accuracy is obtained at the cost of larger models and slower speed of classification.as there are a variety of methods that are available to decrease the size of n-gram model .The experiment with model pruning show that it provides an easy way to balance the size compared the results to the language identifier in Google AJAX Language API, using a subset of 50 languages.

The main task in document processing is text categorization which allows the handling of automated huge streams of documents in electric form main difficulty in handling some classes of documents are the existence of various kinds of textual errors, like spelling and

Grammatical errors in email. Character recognition errors in documents that occur through OCR. These problems shall be handled by the text categorization methods to work correctly for all input levels.

The widely used ranking method has two parameters (maximum n-gram length n and the number of n-grams m) that affect the model. The different experiments have suggested that in case of short test samples, the accuracy for identification can be improved by incrementing the 'm' far beyond the value which were suggested by Canvar and Trenkle in 1994.

This thing can be explained by the fewer number of n-grams in the short input string. Unlike with long inputs, a short text input is not very likely to contain the most frequent n-grams in the language. That is why the model should also contain such words which are not frequently occurring.

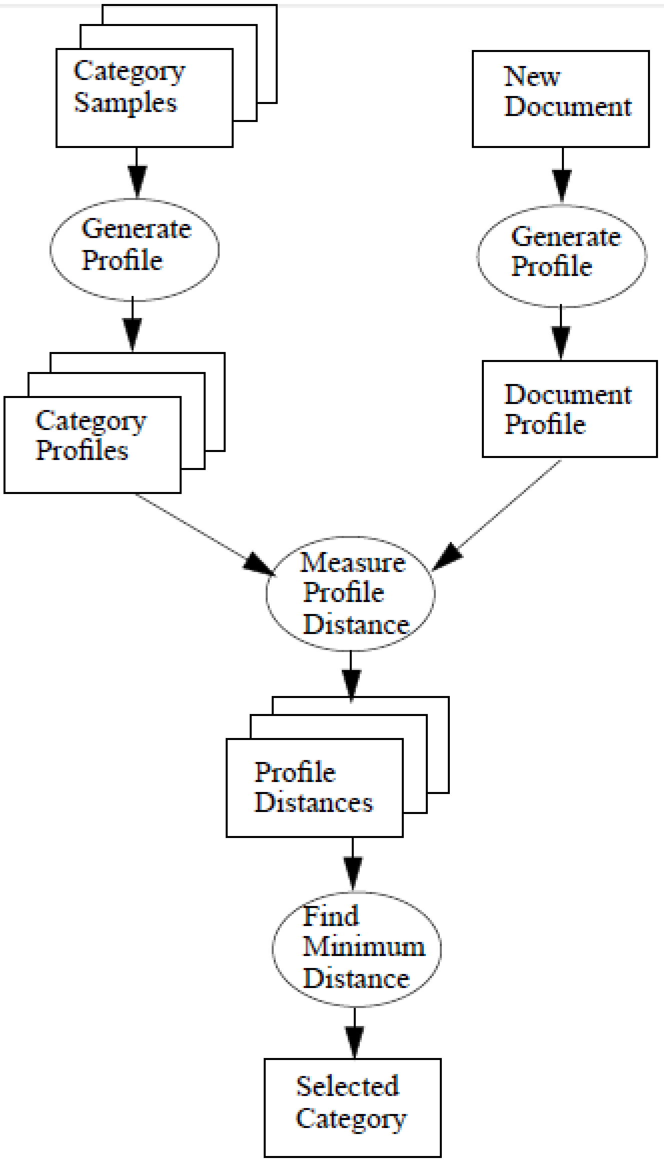
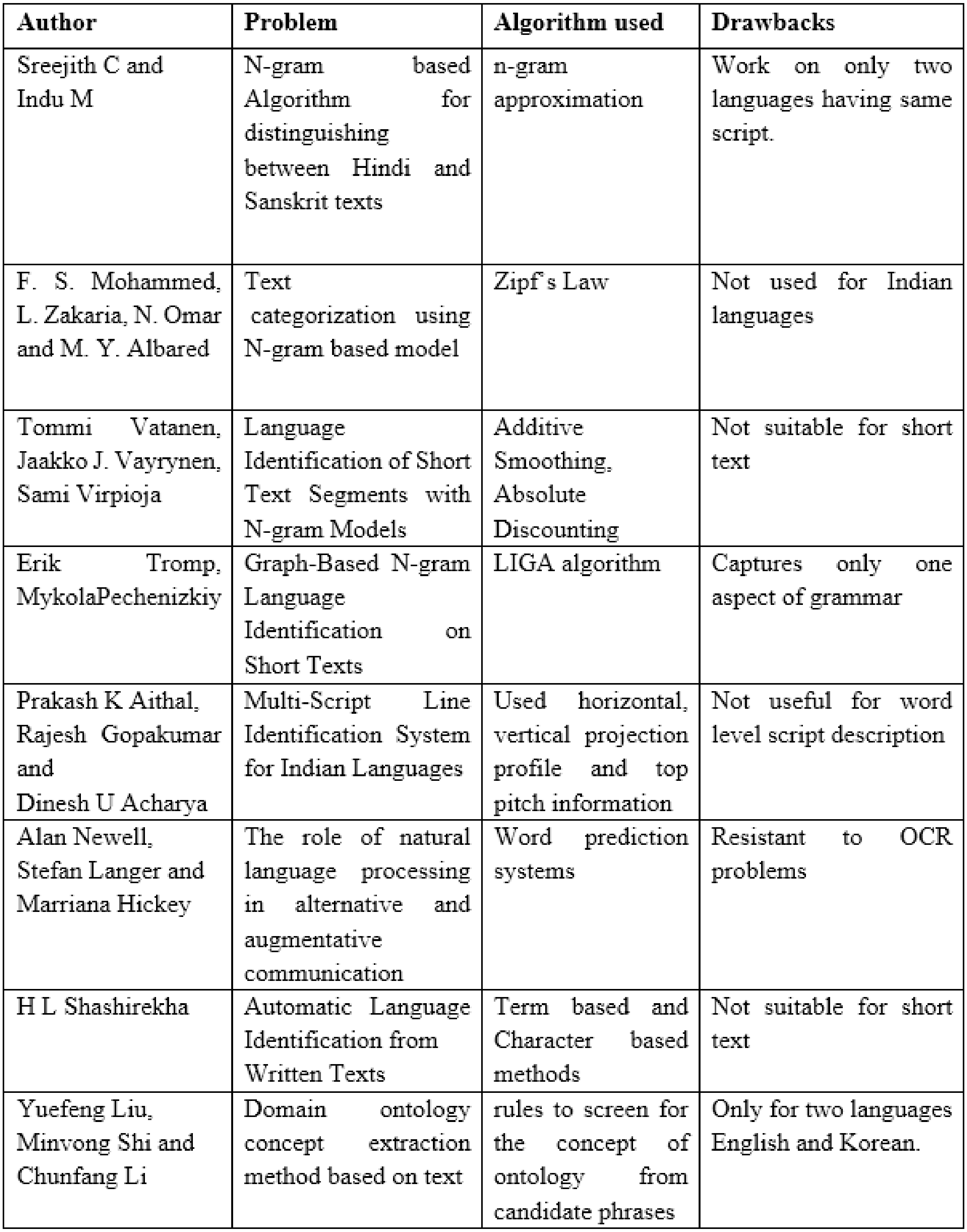


Fig 2.1. shows the basic model for comparison of given text with the corpus.

The literature review that we have done till now is summarised in the following table.

Table.2.1. Literature papers studied.



Author	Problem	Algorithm used	Drawbacks
Sreejith C and Indu M	N-gram based Algorithm for distinguishing between Hindi and Sanskrit texts	n-gram approximation	Work on only two languages having same script.
F. S. Mohammed, L. Zakaria, N. Omar and M. Y. Albared	Text categorization using N-gram based model	Zipf's Law	Not used for Indian languages
Tommi Vatanen, Jaakko J. Vayrynen, Sami Virpioja	Language Identification of Short Text Segments with N-gram Models	Additive Smoothing, Absolute Discounting	Not suitable for short text
Erik Tromp, MykolaPechenizkiy	Graph-Based N-gram Language Identification on Short Texts	LIGA algorithm	Captures only one aspect of grammar
Prakash K Aithal, Rajesh Gopakumar and Dinesh U Acharya	Multi-Script Line Identification System for Indian Languages	Used horizontal, vertical projection profile and top pitch information	Not useful for word level script description
Alan Newell, Stefan Langer and Marriana Hickey	The role of natural language processing in alternative and augmentative communication	Word prediction systems	Resistant to OCR problems
H L Shashirekha	Automatic Language Identification from Written Texts	Term based and Character based methods	Not suitable for short text
Yuefeng Liu, Minvong Shi and Chunfang Li	Domain ontology concept extraction method based on text	rules to screen for the concept of ontology from candidate phrases	Only for two languages English and Korean.

CHAPTER-3 SYSTEM DEVELOPMENT

3.1 N-GRAM

A n-gram is an adjacent grouping of n things from a given arrangement of content or discourse. In the fields of computational semantics and likelihood the things can be phonemes, syllables, letters, words or base sets as indicated by the application. The n-grams are shaped from a string line or from a content corpus. At the point when the things are words, n-grams may likewise be called shingles. N-gram is a n-character part or fundamentally a substring (some portion of a string) of a more drawn out string.

Despite the fact that in the writing the term can incorporate the idea of any co-happening set of characters in a string (e.g., a N-gram comprised of the first and third character of a word), in this paper we utilize the term for adjacent cuts as it were. For all intents and purposes, the string or content is cut into an arrangement of covering N-grams. In our project, we have utilized N-grams of various lengths at the same time. To deal with the circumstances identified with the start and consummation of a string, we have attached clear spaces to the start and completion of the string literals.

3.2 N-GRAM BASED LANGUAGE MODELS

N-gram model is a sort of probabilistic dialect show for anticipating the following thing in a grouping which is a (n-1) arrange Markov demonstrate. In particular, a Language Model (LM) assesses the likelihood of next word from the given words. A N-gram can be essentially characterized as an arrangement of N words. A N-gram dialect show utilizes the historical backdrop of N-1 instantly going before words to register the likelihood (P) of the event of the present word. A N-gram of size 1 is called unigram, measure 2 a bigram (or di gram), estimate 3 a trigram , et cetera.For example, the word

“India is my country" can be decomposed using n gram as:

Character based

1) Unigram characters: (I), (n), (d), (i), (a)...

2) Bigram characters: (In), (nd), (di), (ia)...

3) Trigram characters: (Ind), (ndi), (dia)...

Word based

1) Unigram word: (India), (is), (my), (country)

2) Bigram word: (India is), (is my), (my country)

3) Trigram word: (India is my), (is my country)

The N-gram approximation for calculating the next word
in a sequence is:

$P (X1 \ldots\ldots Xn) = P(X1) P (X2 \mid X1) P (X3 \mid X2) \ldots\ldots P (Xn \mid X1n-1)$

$= \Pi k=1n P (X \mid X1k-1)$

Probability of a complete string:

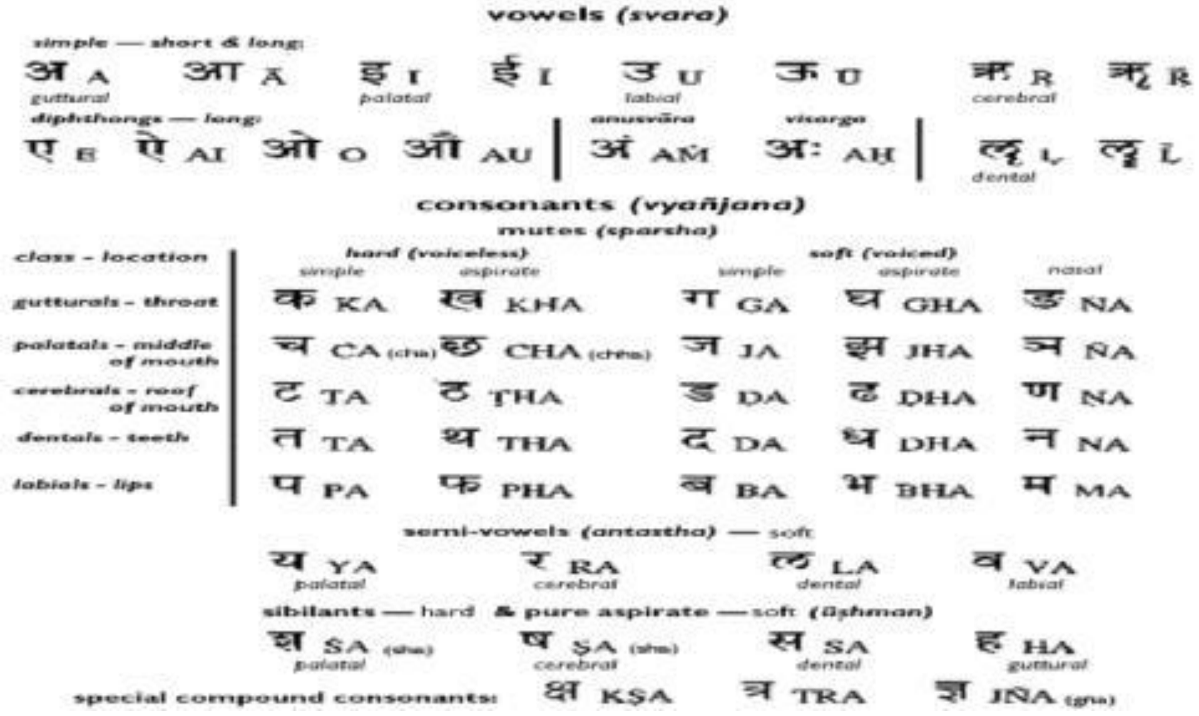
$P(W1n) = P(W1) P (W2 \mid W1) P (W3 \mid W12) \ldots\ldots\ldots P (Wn \mid W1n-1)$

$= \Pi k=1n P (Wk \mid W1k-1)$

Word forecast (speculating the following word from a formerly given information) is the objective ofspelling blunder rectification, programmed discourse acknowledgment, penmanship acknowledgment and so on. In demonstrating of information, the N-gram models have been appeared to be extremely successful and it is a center part in present day measurable dialect applications. Most present-day applications, for example, machine interpretation applications, depend on N-gram based models. N-gram models are generally utilized as a part of measurable regular dialect handling. N-gram circulation is utilized to demonstrate the phonemes (some portion of sound), grouping of phonemes and in discourse acknowledgment. N-grams are likewise essential in common dialect preparing undertakings like grammatical form labelling, regular dialect era, and in addition in applications like origin recognizable proof and slant extraction in prescient content info frameworks for cell phone sets. For dialect ID, successions of characters like (e.g., letters of the alphabet) can be demonstrated for various dialects. In this work, the most successive character and word based 1-grams, 2-grams and 3-grams profiles of Hindi and Sanskrit were utilized for dialect recognizable proof.

3.3 SANSKRIT VS. HINDI

Sanskrit and Hindi, however comparative in script [Fig. 1], indicate critical contrasts regarding their linguistic use and qualities. Both Hindi and Sanskrit has a place with the Aryan gathering of dialects and hence share practically same script. Sanskrit is viewed as the mother dialect of dominant part of Indian dialects, including Hindi, Bengali, Marathi, Oriya, Assamese andGujarati.



vowels (*svara*)

simple — *short & long:*

अ A	आ Ā	इ I	ई Ī	उ U	ऊ Ū	ऋ Ṛ	ॠ Ṝ
guttural		*palatal*		*labial*		*cerebral*	

diphthongs — *long:* ए E ऐ AI ओ O औ AU | *anusvāra* अं AṂ *visarga* अः AḤ | ऌ Ḷ ॡ Ḹ *dental*

consonants (*vyañjana*)

mutes (*sparsha*)

class - location	*hard (voiceless)* *simple*	*aspirate*	*soft (voiced)* *simple*	*aspirate*	*nasal*
gutturals - throat	क KA	ख KHA	ग GA	घ GHA	ङ ṄA
palatals - middle of mouth	च CA (cha)	छ CHA (chha)	ज JA	झ JHA	ञ ÑA
cerebrals - roof of mouth	ट ṬA	ठ ṬHA	ड ḌA	ढ ḌHA	ण ṆA
dentals - teeth	त TA	थ THA	द DA	ध DHA	न NA
labials - lips	प PA	फ PHA	ब BA	भ BHA	म MA

semi-vowels (*antastha*) — soft

य YA	र RA	ल LA	व VA
palatal	*cerebral*	*dental*	*labial*

sibilants — hard & pure aspirate — soft (*ūṣhman*)

श ŚA (sha)	ष ṢA (sha)	स SA	ह HA
palatal	*cerebral*	*dental*	*guttural*

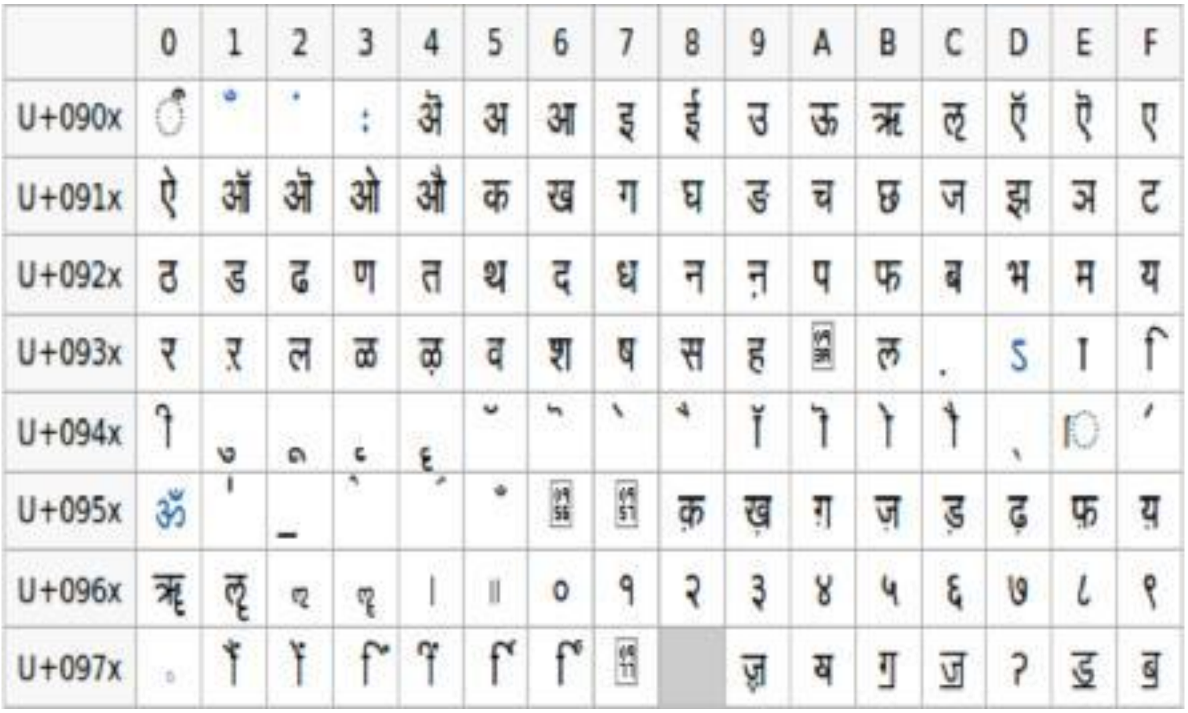
special compound consonants: क्ष KṢA त्र TRA ज्ञ JÑA (gna)

Fig 3.1 Character chart for Hindi.

Also, Sanskrit has its impact on the Dravidian dialects, for example, Telugu, Tamil, Malayalam and Kannada. Hindi is said to have been affected by Sanskrit and furthermore it was created from the other old dialects like Khariboli. Sanskrit is the dialect announced flawlessly fit to be utilized for the PC [15]. Then again, Hindi was not considered so. This is because of the way that Sanskrit syntax is perfect in both phonetics and phonological viewpoints [16].

3.4 DEVANAGRI UNICODE BLOCK

Unicode is a computing industry standard for consistent encoding, portrayal and treatment of content communicated in the vast majority of the world's composition frameworks [19]. Devanagari [Fig.2] is a Unicode square containing characters for composing Hindi and Sanskrit. The range is from 0900 to 097F. The code guides U+0900 toward U+0954 are an immediate duplicate of the characters A0-F4 from the 1988 Indian Standard Code for Information Interchange (ISCII) standard. ISCII is a coding plan for speaking to different written work frameworks of India.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+090x	ऀ	ँ	ं	ः	ऄ	अ	आ	इ	ई	उ	ऊ	ऋ	ऌ	ऍ	ऎ	ए
U+091x	ऐ	ऑ	ऒ	ओ	औ	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट
U+092x	ठ	ड	ढ	ण	त	थ	द	ध	न	ऩ	प	फ	ब	भ	म	य
U+093x	र	ऱ	ल	ळ	ऴ	व	श	ष	स	ह	ऺ	ऻ	़	ऽ	ा	ि
U+094x	ी	ु	ू	ृ	ॄ	ॅ	ॆ	े	ै	ॉ	ॊ	ो	ौ	्	ॎ	ॏ
U+095x	ॐ	॑	॒	॓	॔	ॕ	ॖ	ॗ	क़	ख़	ग़	ज़	ड़	ढ़	फ़	य़
U+096x	ॠ	ॡ	ॢ	ॣ	।	॥	०	१	२	३	४	५	६	७	८	९
U+097x	॰	ॱ	ॲ	ॳ	ॴ	ॵ	ॶ	ॷ		ॹ	ॺ	ॻ	ॼ	ॽ	ॾ	ॿ

Fig 3.2. Devanagari Unicode version 6.1

3.5 THE PROPOSED SYSTEM

In this work, N-grams over Hindi and Sanskrit content information were broke down. The framework is prepared with various N-gram sets of both the dialects . From the preparation set profiles produced, the framework arranges the obscure dialect given in the testing stage by computing the comparability measure. In light of the assessment of various content sources of info, which N-gram profile is dependable for recognizing these two dialects was found. For this, at first, isolate corpora of roughly 1 MB size was made for both Hindi and Sanskrit dialect. This was finished by separating content from

Wikipedia and other online reports. At that point the dialect corpus was sifted and sent as parameter to the preparation profile generator to produce character based and word based unigram, bigram and trigram preparing set for both Hindi and Sanskrit content information. Subsequently, isolate dialect profiles for both dialects were planned in light of N-gram recurrence number from the corpus. Normal dialect Tool Kit (NLTK) in Python dialect was utilized for this experimentation and result definition. Once the preparation set was made, the proposed framework was utilized on irregular test information for characterization and recognizable proof of obscure substance in the advanced online content. This test information was taken from arbitrary reports and messages from Internet with sentences in either Hindi or Sanskrit and the outcomes were noted. For that the test information was separated and sent as parameter to the testing profile generator code to create character based and also word based unigram, bigram and trigram preparing set content information. At that point, the closeness measure was noted down.

3.5.1 Hindi and Sanskrit language profile generation

The corpus was made utilizing the expansive informational collection accessible over the Internet. Once the corpus is framed, a few dialect preparing steps were performed on the gathered information, which includes the accompanying strides:

- Discarding English characters, digits, accentuation and so forth.
- Tokenizing the content into tokens comprising of just words or letters.
- Generating all conceivable character and word based N-grams (for N=1 to 3), and store all N-grams and their number of events (Frequency dissemination).
- Sorting N-grams in view of their frequencies in order.
- Storing N-gram profiles for Hindi and Sanskrit preparing sets independently.

In this work, the initial 30 positioned N-gram words or characters which were observed to be the most generally happening blends in a specific dialect, was utilized to make the preparation profile for that dialect. Most composition frameworks bolster more than one dialect. The Cyrillic scripts utilized by about the majority of the dialects from the Soviet Union, is a case of multi-dialect supporting frameworks. Given a content that uses a specific composition framework, it is important to decide the dialect in which it is composed before further preparing is conceivable. There are a few expansive ways to deal with the dialect characterization issue. One evident procedure is to keep a vocabulary for every conceivable dialect, and afterward to look into each word in the specimen content to find in which dictionary it falls. N-gram recurrence profile creation for Hindi and Sanskrit dialects are produced as appeared in Fig.3.

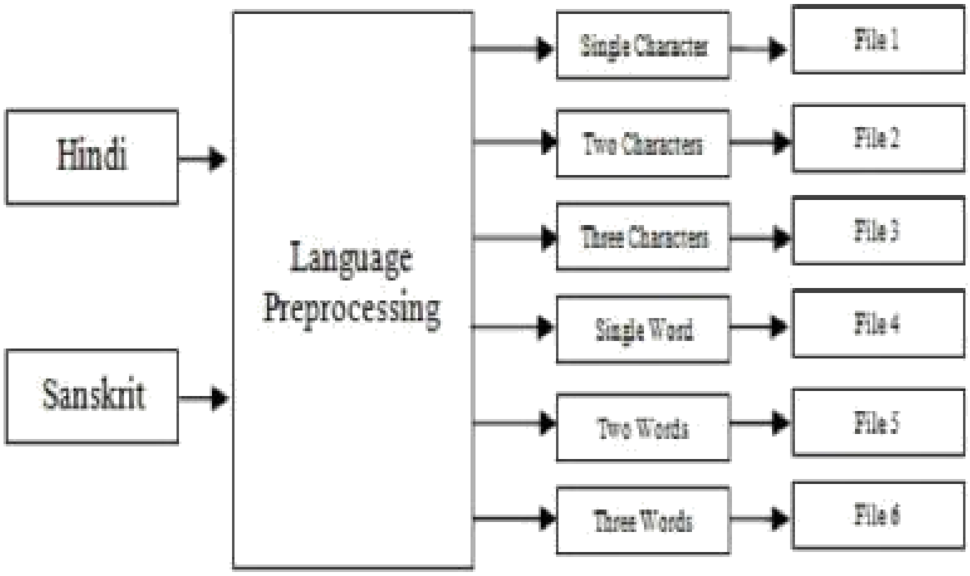


Fig 3.3. N-gram frequency profiles creation for Hindi and Sanskrit languages

3.5.2 Test data profile generation

For an arbitrary content given, the dialect to which it belongs needs with be found. With a specific end goal to recognize this, a character based and word based unigram, bigram and trigram testing profile was produced. This testing profile was contrasted and each of the prepared profiles produced for Hindi and Sanskrit dialects for measuring the likeness between them. For each word or character in the N-gram test profile, the preparation set was scanned for the nearness of a similar word or character. At first, the comparability number is set to none and later when a match is found, the similitude check is increased by one or else it is rejected. This is rehashed for all N-gram profiles in the testing profile. The whole of the closeness check estimations of every N-gram profile when contrasted and the comparing N-gram preparing profile of every dialect is recorded.

This is known as the likeness measure. Presently, for every N-gram profile in the testing profile, the greatest comparability measure is found. The dialect of the preparation profile with most extreme esteem is distinguished as the dialect of the test information [Fig. 4]. The past strides were rehashed for various test information of Hindi and Sanskrit.

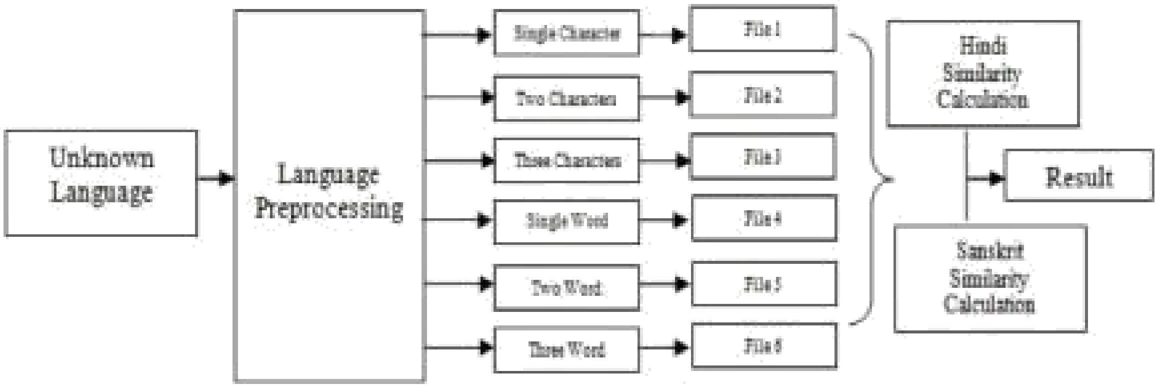


Fig3.4. N-gram frequency profiles creation for test data and similarity measurement

3.6 PYTHON 3.0

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its outline reasoning focuses on code coherence, and its linguistic structure enables developers to express ideas in less lines of code than conceivable in other environments, for example, C++ or Java. The language gives builds planned to empower composing clear projects on both a little and substantial scale.

Python supports various programming models, including OOPS, basic and utilitarian programming or procedural styles. It includes a dynamic framework and automatic memory management t and garbage collection and has a substantial and thorough standard library.

Python interpreters are 4available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Python has a substantial standard library ordinarily referred to as one of Python's most noteworthy qualities, giving instruments suited to many undertakings. This is considered and has been depicted as a batteries incorporate Python rationality. For Internetconfronting applications, numerous standard arrangements and conventions, (for example, MIME and HTTP) are upheld. Modules for making graphical UIs interfacing with social databases, pseudorandom number generators, math with discretionary exactness decimals, controlling standard expression, and doing unit testing are likewise included.

A few sections of the standard library are covered by regulations and specifications (for instance, the Web Server Gateway Interface (WSGI) usage wsgiref takes after PEP 333), yet most modules are definitely not. They are indicated by their code, documentation, and test suite (if provided). In any case, in light of the fact that the vast majority of the standard library is cross-stage Python code, just a couple of modules need modifying or reworking for variation executions.

The standard library is not expected to run Python or insert it in an application. For instance, Blender 2.49 overlooks the greater part of the standard library.

The Python Package Index, the official storehouse containing outsider programming for Python, contains more than 92,000 bundles offering an extensive variety of usefulness, including:

- graphical UIs, web structures, multimedia, databases, systems administration and interchanges
- test structures, mechanization and web scratching, documentation apparatuses, framework organization
- scientific figuring, content handling, picture preparing

3.7 NATURAL LANGUAGE TOOL KIT (NLTK)

NLTK is a main stage for building Python projects to work with human language information. It gives simple to-utilize interfaces to more than 50 corpora and lexical assets, for example, WordNet, alongside a suite of content preparing libraries for order, tokenization, stemming, labelling, parsing, and semantic thinking, wrappers for modern quality NLP libraries, and a dynamic exchange gathering.

On account of a hands-on guide presenting programming basics close by points in computational phonetics, in addition to far reaching API documentation, NLTK is reasonable for language specialists, engineers, understudies, teachers, scientists, and industry clients alike. NLTK is accessible for Windows, Mac OS X, and Linux. The best part is that NLTK is a free, open source, group driven venture.

NLTK has been called "a superb apparatus for educating, and working in, computational semantics utilizing Python," and "a stunning library to play with natural language."

Natural Language Processing with Python gives a pragmatic prologue to programming for dialect processing. Composed by the makers of NLTK, it directs the reader through the essentials of composing Python programs, working with corpora, arranging content, examining semantic structure, and then some.

CHAPTER-4 PERFORMANCE ANALYSIS

A similar content arrangement approach effectively stretches out to the idea of utilizing N-gram recurrence to gauge subject similitude for records that are in a similar dialect. Surely, the approach reaches out to a multi-dialect database where both the dialect and the substance of the report are of enthusiasm for the recovery procedure. To distinguish the proper newsgroup for newsgroup articles, we have utilized grouping framework so as to test this approach. The articles for this examination originated from a portion of the Usenet newsgroups. We wished to perceive how precisely the framework would recognize which newsgroup each message initially originated from. The characterization technique was as per the following:

- Obtained preparing sets for each newsgroup. To finish this reason, we have utilized the oftentimes made inquiries article records. Numerous newsgroups consistently distribute such FAQs as a method for diminishing activity in the gathering by noting questions or talking about issues that surfaced a great deal in the gathering.
- Hence the FAQ for a newsgroup tries to characterize what he newsgroup is about and characterizes a great deal more central innovation for the gathering. The FAQs we have gathered are in the vicinity of 18K and 132K long. The FAQ ought to need to give satisfactory covering of topic of the newsgroup notwithstanding of no specific organization necessity.
- Computed N-gram frequencies on the newsgroup's FAQ. These are precisely the same as alternate sorts of N-gram recurrence profiles specified before. The subsequent profiles are very little, on the request of 10K bytes or less.
- Computed an article's N-gram profile in a manner like that for registering the profile for every FAQ. The articles found the middle value of 2K long and the subsequent article profiles were on the request of 4K long .
- Computed a general separation measure between the article's profile and the profile for each newsgroup's FAQ. The FAQ profile with the littlest separation measure from the article's profile figured out which newsgroup to group the example as.
- Compared the chose newsgroup from the real one the article originated from.

In past a few years a considerable measure of research has been done in the area of information preparing and multilingual literary information. This is for a few reasons: a developing accumulation of arranged and all around appropriated information, the improvement of correspondence framework and the Internet, the expansion in the quantity of individuals associated with the worldwide system and whose first language is not English. This has made a need to arrange and prepare colossal volumes of information. It is an exorbitant in term of time and for work force. They are unbendable and speculations to different territories are for all intents and purposes unimaginable , so we attempt to create programmed techniques to auto recognize the printed dialect.

Electronic reports originated from a wide assortment of sources. Many are created with different word preparing programming bundles, and are subjected to different sorts of programmed investigation, e.g., spelling checkers, and also to manual altering and correction. Numerous different records, be that as it may, don't have the advantage of this sort of examination, and along these lines may contain critical quantities of blunders of different sorts. Email messages and notice load up postings, for instance, are frequently formed on the fly and sent without even the most superficial levels of investigation and adjustment. The paper records what are for the most part carefully examined and went and go through an OCR framework will no questioned do contain in any event some level of acknowledgment blunders. It is accurately on these sorts of archives, where assist manual review and adjustment is troublesome and expensive, that there would be the best advantage in programmed preparing. Content classification is one of the crucial sorts of archive preparing, in which an approaching report is appointed to some previous class.

One of the uses of such framework is to course news articles from a newswire. Another application would deal with digitized paper documents.

Content order is one basic sort of archive preparing, in which an approaching record is doled out to some previous classification. One of the use of such framework is to course news articles from a newswire. Another application would deal with digitized paper archives.

These applications have the accompanying qualities:

- The arrangement must work dependably disregarding literary mistakes.

- The arrangement must be productive, expending as meager stockpiling and preparing time as could be allowed, on account of the sheer volume of archives to be dealt with.
- At the point when a given record does not coordinate any class, or when it falls between two classifications, the order must have the capacity to remember it. This is so in light of the fact that the class limits are not obvious.

Dialect Identification is the undertaking of consequently recognizing the language(s) in which the substance is composed in an archive (website page, content report). Because of the across the board of the web, distinguishing proof of dialects has turned into an essential pre-handling venture for various applications. These applications incorporate machine interpretation, Part-of-Speech labelling, etymological corpus creation, supporting low-thickness dialects, availability of online networking or client produced content, web indexes and data extraction notwithstanding preparing multilingual records. To connect the advanced separation between the diverse dialect clients in a multilingual nation like India, LI can be utilized as a compelling apparatus. In this paper analyst has introduced a short outline of the difficulties required in programmed dialect recognizable proof. The essayist has additionally specified about the current philosophies and a portion of the instruments accessible for dialect distinguishing proof.

The dialect recognizable proof is a fascinating issue in its own. In any case, LI can be viewed as a stage of the huge procedure, all things considered, critical thinking process. Exact LI can encourage utilization of foundation data about the dialect and utilization of more specific methodologies in numerous characteristic dialect preparing errands managing an accumulation or a flood of writings, each of which can be composed in an alternate dialect.

The multilingual assessment examination via web-based networking media would be an ordinary rousing case. The examination paper introduced by Tromp in 2011 demonstrates a broad test learn about the multilingual assumption characterization that can be performed all the more precisely when the procedure is part into four stages; LI grammatical form labelling, subjectivity identification and extremity location. This ends up noticeably conceivable on the grounds that at each progression after LI, models that use dialect particular information can be connected. Clearly, if dialect of some content is recognized erroneously then this blunder will impact the prospective strides of the multilingual assessment examination and consequently bargain the utilization of the moderately confused four stage methodology. Along these lines, it is very alluring to limit the blunder of LI.

4.1 SUBJECT CLASSIFICATION TEST DATA

To test this framework, we gathered article tests from five Usenet newsgroups. These newsgroups are appeared . We picked these five since they were all subfields of software engineering, and in this way, would give a chance to testing how the framework may befuddle newsgroups that were fairly firmly related. The article extraction program additionally expelled the typical header data, for example, subject and watchword distinguishing proof, leaving just the body of the article. This kept any matches that were too firmly affected by standard header data for the newsgroup (e.g., the newsgroup name). For the profiles , we picked the FAQs appeared in Table 5. Here we need to notice that there is some level of cover with the chose newsgroups for the came about analysis: -

- There are FAQs for rec.games.go and comp .robotics, but no articles from either group.
- There are two FAQs related to compression, covering slightly different areas.
- There are some articles present for computer graphics, but having no FAQs.

Given this setup, we ran the classification procedure outlined above for all 778 newsgroup articles against the 7 selected FAQs. Our results are shown in Table 6.

The following Results can be seen in the following tables:-

- The security FAQ provides 77% coverage of alt security.
- The compilers FAQ provides 80% coverage of comp.compilers.
- The jpeg and the compression FAQs together provides 75% coverage of computer compression.
- The go FAQ picked up only 3 articles altogether, indicating that its coverage is almost completely disjoint from the five selected newsgroups. There are also existing somewhat of weaker results:

- The FAQ related to robotics has picked up 11 artificial intelligence articles and 25 graphical articles. This is so because of the relative proximity of these fields to robotics.
- There is only 30% coverage of the computer artificial intelligence group provided by the FAQ.

Seeing that the computerized reasoning FAQ is almost twice as substantial as the following biggest FAQ, we can theorize that it might in actuality cover excessively material, therefore diverting from the factual way of the N-gram recurrence measure . This may likewise mirror the way that comp.ai truly comprises of a few related however particular subgroups (master frameworks, connectionism/neural systems, vision frameworks , hypothesis provers, and so forth.) that happen to have the same newsgroup.

The articles from PC design were conveyed among alternate FAQs. This is normal since we did exclude the FAQ from PC design for the articles to be coordinated. It is the indicate take note of that the most grounded coordinating FAQ for these articles was the jpeg compression. It covers a pressure standard for graphical data. That is the reason it was a solid and sensible contender for the match. It earned a 44% scope of PC representation.

By and large, the framework works great given the to some degree Overall, the framework works great given the to some degree uproarious nature of the newsgroups, and the fundamentally fragmented nature of the FAQ records. In spite of the fact that we don't dissect it here, careless manual examination of the outcomes demonstrated that when the framework coordinated an article against the off base FAQ, the right FAQ was by and large the second decision. Something else to remember is that we didn't decide the genuine substance of each article to check whether it properly had a place with the gathering it showed up in. In Usenet newsgroups, spurious cross-posting of immaterial articles (e.g.,)

4.2 ARTICLE SAMPLES

In past a few years a great deal of research has been done in the range of information preparing and multilingual literary information. This is for a few reasons: a developing gathering of organized and all around appropriated information, the improvement of correspondence framework and the Internet , the expansion in the quantity of individuals associated with the worldwide system and whose native language is not English. This has made a need to arrange and prepare immense volumes of information. It is an expensive in term of time and for some personnel They are unyielding and speculation to different zones are basically unthinkable, so we attempt to create programmed strategies to auto distinguish the printed dialect .

Electronic reports originated from a wide assortment of sources. Many are produced with different word preparing programming bundles, and are subjected to different sorts of programmed investigation, e.g., spelling checkers, and to manual altering and update. Numerous different reports, be that as it may, don't have the advantage of this sort of investigation, and in this way may contain noteworthy quantities of mistakes of different sorts. Email messages and notice load up postings, for instance, are regularly formed on the fly and sent without even the most quick levels of investigation and amendment. The paper archives what are by and large carefully checked and went and go through an OCR framework will no doubt do contain at any rate some level of acknowledgment blunders. It is decisively on these sorts of records, where encourage manual review and adjustment is troublesome and exorbitant, that there would be the best advantage in programmed preparing. Content arrangement is one of the central sort of report handling, in which an approaching record is allotted to some prior classification. One of the use of such framework is to course news articles from a newswire. Another application would deal with digitized paper files .

The related types of the words, for example, – progress, progressed, progressing and headway and so on are having a considerable measure in like manner when seen as set of n-grams. Keeping in mind the end goal to get equal outcomes with all words, the framework should play out the word stemming (i.e.) the apportioning of sound, which would require that the framework ought to have profound comprehension of the dialect in which the report is composed.

The N-gram recurrence approach gives dialect autonomy to free. Another preferred standpoint of this approach is the capacity to work similarly well with short and long archive. Different applications are the insignificant stockpiling and computational necessities .
We have avoided making any earlier supposition on our corpus in the tests. In genuine word applications with particular dialect sets, dialect particular data can be misused , e.g., in pre-handling.

For example, on the off chance that one works with dialects plainly comprising of words, joining word-based dialect models with character-based models could give enhanced exactness.

The generally utilized positioning strategy has two parameters (greatest n-gram length n and the quantity of n-grams m) that influence the model. The diverse trials have proposed that in the event of short test tests, the precision for distinguishing proof can be enhanced by increasing the "m" a long ways past the esteem which was recommended by Canvar and Trenkle in 1994.

This thing can be clarified by the less number of n-grams in the short info string. Not at all like with long data sources, a short content info is not prone to contain the most continuous n-grams in the dialect. That is the reason the model ought to likewise contain such words which are not often happening. Or maybe shockingly, adjusted KN has the most minimal distinguishing proof precision of the progressed smoothing techniques. In any case, by figuring normal perplexities for the test information of a similar dialect the model was prepared, we found that the adjusted KN enhanced the forecasts over the other smoothing techniques. This proposes the execution of the dialect display all things considered does not really mirror the order exactness in dialect ID with the guileless Bayes classifier. As indicated by the papers composed by "Goodman" in 2001, Combining distinctive sorts of dialect models is one strategy for enhancing the forecast capacity. We made two starting analyses with direct mix.

In reverse n-gram models (Duchateau et al., 2002), where the probabilities of the characters are evaluated adapted on the accompanying characters, enhanced the distinguishing proof exactness 0.9% total with 5-gram models smoothed with supreme reducing. There could be another straightforward approach to enhance the outcomes by incorporating a model in foundation that is prepared by the full information of different dialects. The another basic method for enhancing the outcomes may infuse or embeddings with a foundation demonstrate prepared with the full informational index for different dialects that bolster multilingual properties . As clarified by Zhai and Lafferty (2001), the foundation model ought to decrease the impact of normal characters, like the backwards record recurrence weighting connected in vector space models . For future work there are more broad examinations with model pruning is cleared out. Considering the sizes of the n-gram models, there are a few strategies for making more reduced models that are not detailed previously. We made beginning tests with variable length n-gram models prepared with the developing calculation by Siivola et al. (2007), yet the models did not enhance the consequences of 5-gram models (as did not full models utilizing longer n-grams). Be that as it may, we didn't take a stab at utilizing check shorts or grouping , which are accounted for to be more proficient than pruning alone in word-based n-gram models (Goodman and Gao, 2000). In general, these strategies ought to be more valuable with bigger preparing corpora.

Language Identification might be considered as an uncommon instance of multi name content order with a predefined set of names speaking to the dialects of the records in the preparation set and LI as the errand of relegating a subset of the predefined marks (dialects) to a content archive under thought. For the grouping of the multi lingual reports, this situation holds great. Be that as it may, for LI of monolingual archives, multiclass content classifiers with "k" classes can be considered relying on the quantity of dialects to be distinguished. Many machine learning and factual methodologies in blend with phonetic methodologies (for highlight extraction) are investigated by numerous scientists to deliver the issues identified with the ID of their local dialects and dialects in their neighbouring areas. A short review of a portion of the well-known calculations is given beneath:

A little, quick and vigorous N-gram based strategy proposed by W. B. Cavnar and J. M. Trenkle [1] for content order has been connected by numerous scientists effectively for LI [3, 8, 9, 10, 11, 13]. N-gram is a N-character cut of a more drawn out string and N-grams of various lengths will be utilized. The general casing work of a N-gram based technique is to process the dialect profile for every dialect in the preparation set and the objective profile for each test report under thought. The proposed framework initially figures the separation measure between the objective profile and other dialect profile set in the preparation informational index. The framework then figure the separation to choose the dialect with having less separation when contrasted and target profile information. For Language recognizable proof, a few specialists have utilized the varieties of n-gram based techniques as an apparatus . Specially appointed aggregate recurrence increases of n-grams to distinguish the short content of twelve dialects have been utilized by Bashir Ahmed et. [3]. They have proclaimed that the Naïve Bayes technique is practically identical with the speed of their strategy and the precision is tantamount to the rank-arrange measurement technique. Erik Tromp and Mykola Pechenizkiy [8] propose a diagram based N-gram approach for the distinguishing proof of dialects in generally short and sick composed writings. This approach offers significance to word requesting alongside word event spoken to as a chart, when contrasted with a large portion of the methodologies which offers significance to few words as it were. The analysts probed the gathered arrangement of data assembled from Twitter written in six unique dialects and established that their technique was surprisingly more exact and exact than the current N-gram based methodologies and less experienced overfitted and area specified phrases and figures of speech.

Some work on the auto-recognizable proof of the dialects in the site pages have been accounted for by the writing audit.

The reports some works on the identification of languages in web pages. Ali Selamat [9] has proposed an improved and efficient approach based on the combination of the existing n-gram approach and the modified n-gram approach. To identify 12 languages written in Roman and Arabic scripts, they have selected the features based on distance measurement from the original n-grams approach and features based on a Boolean matching rate. For the identification of Roman and Arabic script languages n-gram based approach was able to improve the identification of the content, which was shown by the results. Yew Choong Chew et al., [10] has proposed an improved n-gram based algorithm for LI of web pages for Asian languages based on non-Latin script. The algorithm's performance was evaluated on the basis of written text corpus of 1,660 web pages. These web pages were gathered from 182 languages from Asia, Africa, America, Europe and Oceania. The algorithm has achieved an accuracy rate of 94%. Since A lot of work has been dine in the recognition of various languages, but still there is less workreported for the automatic identification of the regional languages in India.

An N-gram based algorithm for the identification of documents with Kannada, Telugu and English sentences by processing n-gram of only the last word of the sentence instead of complete sentence was proposed by Deepmala and Ramakanth Kumar [11]. They have used it as a pre-processing step for the detection of sentence boundaries and found encouraging results. There are some challenges of its own are present in the identification of the languages which shares common scripts. An N-gram based algorithm to distinguishing between Hindi and Sanskrit texts, which are having a common script, was proposed by the Sreejith C. [12]. They have achieved an accuracy of 97% by using the unigram, bigram and trigram based training data set profile. Kavi Narayana Murthy and G. Bharadwaja Kumar [5] formulate LI as a two class pair wise classification problem using Multiple Linear Regression (MLR) for the classification of small text samples of Indian languages. This paper also illustrates the issues related to the identification of scripting languages and Indian languages.

A Roli framework was proposed by Kosuru Pavan et al., [6] to address the difficulties in the programmed distinguishing proof of the Romanized content. Roli is a N-gram based approach. Rolihas likewise make utilization of sound based likeness of words. Roli has accomplished a normal exactness of 98% regardless of the spelling minor departure from five Indian dialects:

- Hindi
- Telugu
- Tamil
- Kannada
- Malayalam

To distinguish the dialect in the multilingual content information, a cross breed calculation gotten from the blend of K-means and the simulated subterranean insect class calculations have been proposed by Abdelmalek Amine et. al., [7]. They work on the N-grams properties of the characters. They group together comparable messages and find the quantity of dialects in a completely unsupervised way. Anidentification procedure in view of etymological components called as shut word classes which incorporates Adverbs, Articles, Conjunctions, Interjections, Numerals, Prepositions and Pronouns was proposed by Rafael DueireLins and Paulo Gonçalves Jr. [2]. They have probed four dialects to be specific Portuguese, Spanish, French and English. To recognize the dialect of a given Web report some new similarith measures and heuristics have been talked about by Bruno Martins and Mário J. Silva [4] utilizing n-gram based calculations.

The calculation was used as a component of Portuguese web index (www.tumba.pt) and was developed from 25 diverse dialect's information accumulated from the newsgroups and the Web.MarcosZampieri [13] has proposed straightforward sack of-words approach for distinguishing proof of dialect assortments and has performed tests utilizing Multinomial Naïve Bayes (MNB), Support Vector Machines (SVM) and J48 classifier. The outcomes demonstrate that their technique has execution equivalent to cutting edge strategies in light of n-gram models. MarcoLui et al, [15] presents a framework for dialect recognizable proof in multilingual archives utilizing a generative blend demonstrate propelled by directed theme displaying calculations, joined with a report portrayal for monolingual records. The outcomes shows that the proposed framework outflanks elective methodologies from the writing on manufactured information, and in addition on genuine information from related research on etymological corpus creation for lowdensity dialects utilizing the web as an asset.

Distinguishing dialects from boisterous content is a genuine test in LI as most techniques work on clean messages as well as long messages, however frequently introduce a disappointment when the content is defiled or too short. Kheireddine Abainia et al., [14] have proposed a half and half approach for the recognizable proof of dialects of loud short messages and probed the gathering of writings from a few exchange discussions containing a few sorts of clamors relating to 32 dialects. Their half approach which is characterized as a mix of term-based and character-based strategies are very intriguing and present great dialect distinguishing proof exhibitions in uproarious writings.

CHAPTER-5 CONCLUSIONS

The writing review gives a short diagram of the difficulties required in programmed LI, existing approaches and a portion of the apparatuses accessible for programmed LI. It can be watched that the majority of the strategies utilize N-gram model or variety of N-gram demonstrate in blend with different procedures for highlight extraction and afterward utilize machine learning methods for the distinguishing proof of dialects. Because of an upsurge in the quantity of web journals, sites and electronic stockpiling of literary information, the business significance of programmed content order applications has expanded and much research is as of now centred around there. Content characterization can be computerized effectively utilizing machine learning methods, however pre-preparing and include choice strides assume a critical part in the size and nature of preparing information given to the classifier, which thus influences the classifier precision. Refined content classifiers are not yet accessible for a few territorial dialects, which if created would be valuable for a few administrative and business ventures. Incremental content arrangement, multi-subject content characterization, finding the nearness and logical utilization of recently advancing terms on online journals and so forth are a portion of the territories where future research in programmed content grouping can be coordinated.

In writing overview we have concentrated the issue of dialect recognizable proof on moderately short messages regular for online networking like Twitter. Prior chips away at dialect ID indicated promising and very exact outcomes for all around developed, adequately sufficiently long messages. In any case, the outcomes were appeared to fall apart significantly when writings turn into a couple of hundred characters in length. The N-gram recurrence strategy gives an economical and profoundly viable method for ordering archives. It does as such by utilizing tests of the coveted classes instead of falling back on more muddled and exorbitant strategies, for example, common dialect parsing or collecting nitty gritty vocabularies. Basically this approach characterizes an "order by case" technique. Gathering tests and building profiles can even be dealt with in a to a great extent programmed way. Likewise, this framework is impervious to different OCR issues, since it relies on upon the measurable properties of N-gram events and not on a specific event of a word. In spite of the fact that the current framework as of now has shown great execution, there is extensive space for further work.

Albeit numerous dialect distinguishing proof strategies work extremely well for archives or other long messages, paper audit affirmed that the recognizable proof of short content sections was not a tackled issue. The precision of the examined strategies was found to diminish fundamentally when the recognized content gets shorter. Along these lines, finding the best techniques and advancing the parameters is critical with short content sources of info.

5.1 FUTURE SCOPE

From writing thestudy, we have found that the current framework as of now has exhibited great execution, there is impressive space for further work.

Right now, the framework utilizes various diverse N-grams, some of which at last are more reliant on the dialect of the report than the words containing its substance. By discarding the insights for those N-grams which are to a great degree basic since they are basically components of the dialect, it might be conceivable to show signs of improvement segregation from those measurements that remain. It is likewise conceivable that the framework ought to incorporate some extra measurements for rarer N-grams, accordingly increasing further scope.

This framework as of now handles just dialects that are straightforwardly representable in ASCII. The rising ISO-6048/UNICODE standard opens up the likelihood of applying the N-gram recurrence thought to the majority of the dialects of the world, including the demographic ones.

Till now we have done word based distinguishing proof of content. In further work we'll be doing character based recognizable proof of content , to recognize dialects at more profound level or base level.

In further work, we will look at how the technique performs on different dialects. We will examine other Indian dialects which are especially talked in an area as it were. We will likewise explore the impact of the quantity of writings in the corpus.

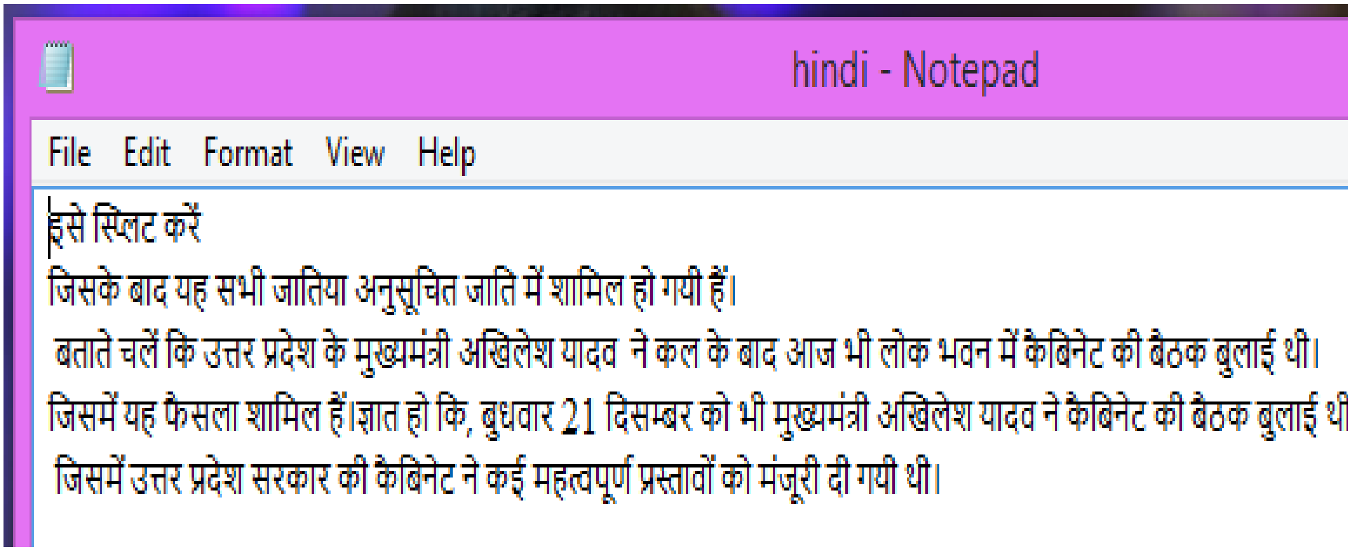
5.2 APPLCATIONS

Auto identification of text can be useful in-

- Automatic summarization
- Social media text categorization.
- Library Management System.

5.3 DEMO

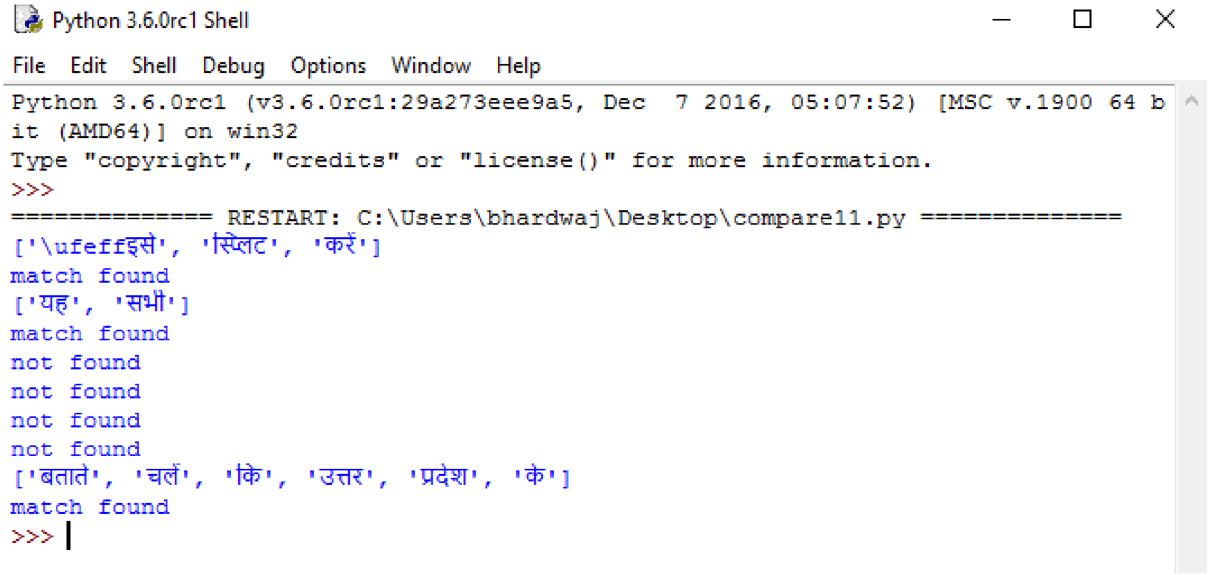
Text to be checked:



इसे स्प्लिट करें
जिसके बाद यह सभी जातिया अनुसूचित जाति में शामिल हो गयी हैं।
बताते चलें कि उत्तर प्रदेश के मुख्यमंत्री अखिलेश यादव ने कल के बाद आज भी लोक भवन में केबिनेट की बैठक बुलाई थी।
जिसमें यह फैसला शामिल हैं।ज्ञात हो कि, बुधवार 21 दिसम्बर को भी मुख्यमंत्री अखिलेश यादव ने केबिनेट की बैठक बुलाई थ
जिसमें उत्तर प्रदेश सरकार की केबिनेट ने कई महत्वपूर्ण प्रस्तावों को मंजूरी दी गयी थी।

Fig 5.1. Hindi text data sample.

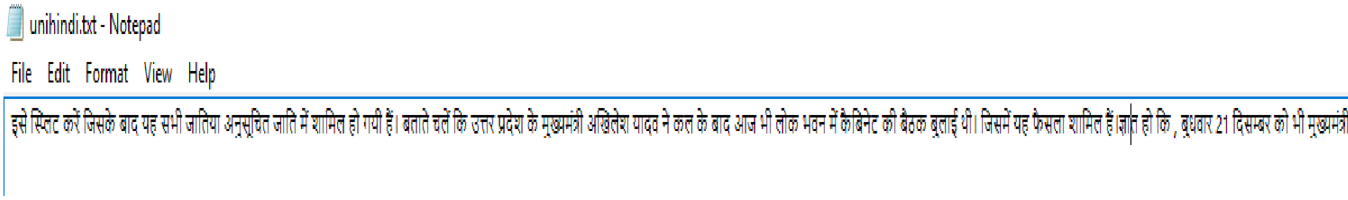
Python shell output during comparison.



```
Python 3.6.0rc1 (v3.6.0rc1:29a273eee9a5, Dec  7 2016, 05:07:52) [MSC v.1900 64 b
it (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\bhardwaj\Desktop\compare11.py ==============
['\ufeffइसे', 'स्प्लिट', 'करें']
match found
['यह', 'सभी']
match found
not found
not found
not found
not found
['बताते', 'चलें', 'कि', 'उत्तर', 'प्रदेश', 'के']
match found
>>> 
```

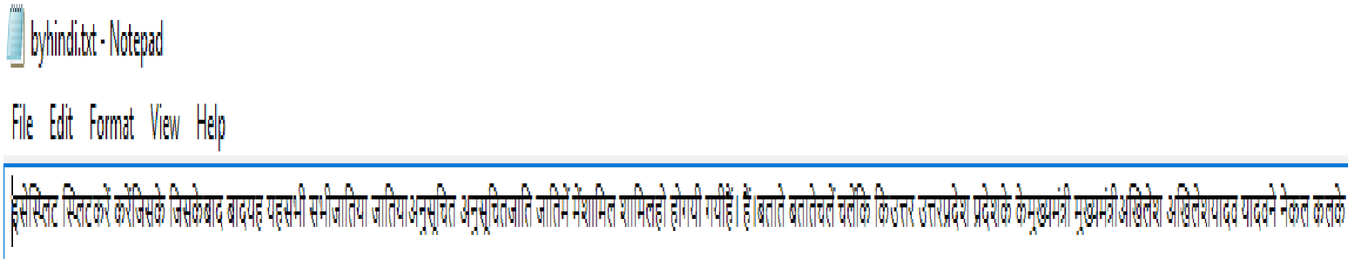
Fig 5.2 Python shell showing word match found

Results achieved after the running of code files:



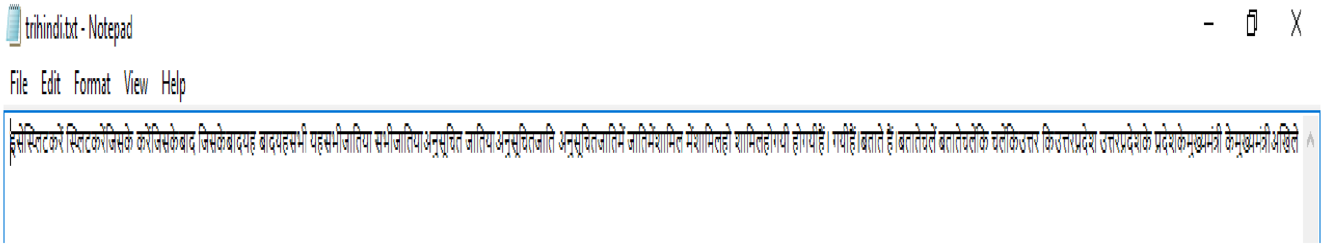
इसे स्प्लिट करें जिसके बाद यह सभी जातिया अनुसूचित जाति में शामिल हो गयी हैं। बताते चलें कि उत्तर प्रदेश के मुख्यमंत्री अखिलेश यादव ने कल के बाद आज भी लोक भवन में कैबिनेट की बैठक बुलाई थी। जिसमें यह फैसला शामिल हैं।ज्ञात हो कि , बुधवार 21 दिसम्बर को भी मुख्यमंत्री

Fig 5.3 Unigram word file for Hindi text



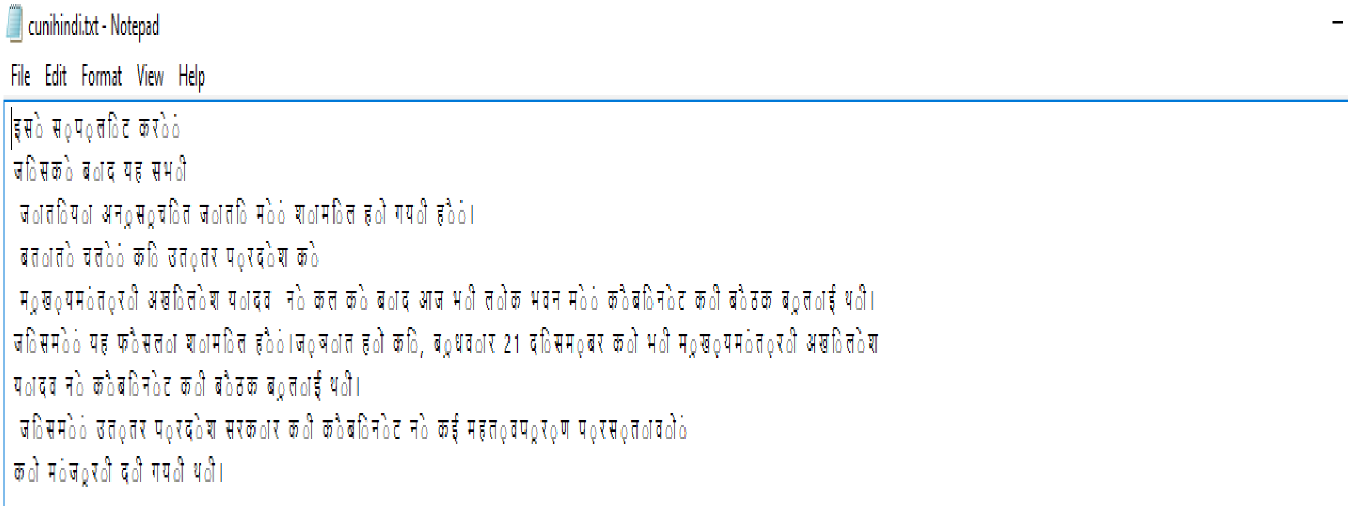
इसेस्प्लिट स्प्लिटकरें करेंजिसके जिसकेबाद बादयह यहसभी सभीजातिया जातियाअनुसूचित अनुसूचितजाति जातिमें मेंशामित शामिलहो होगयी गयीहैं। हैं। बताते बतातेचलें चलेंकि किउत्तर उत्तरप्रदेश प्रदेशके केमुख्यमंत्री मुख्यमंत्रीअखिलेश अखिलेशयादव यादवने नेकल कलके

Fig 5.4 Bigram file for Hindi text



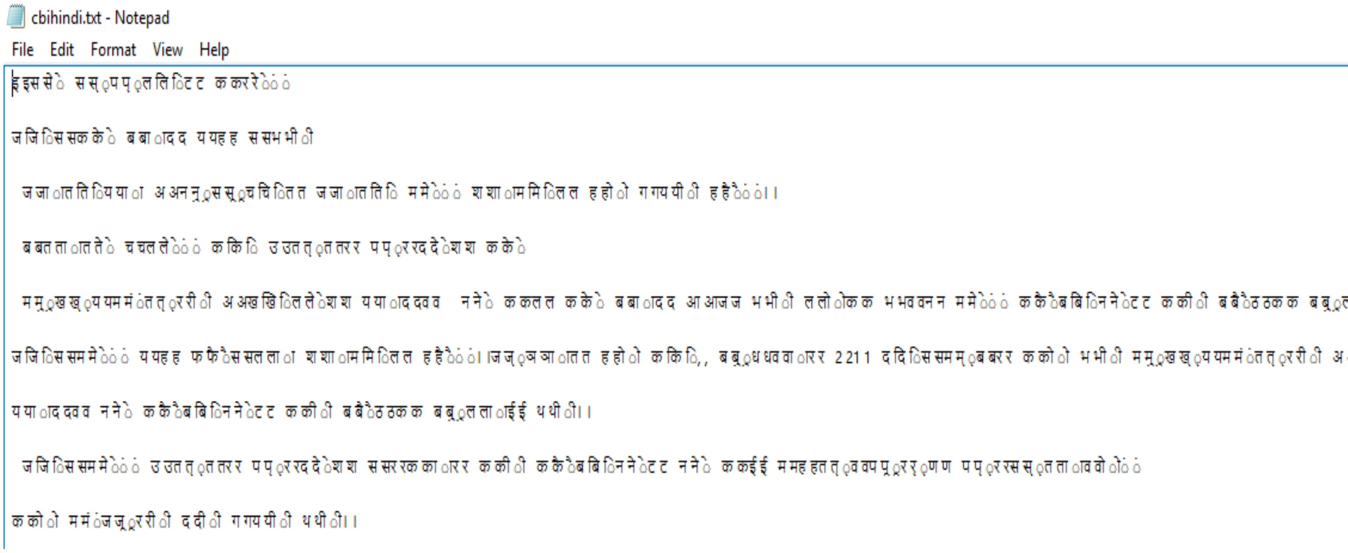
इसेस्प्लिटकरें स्प्लिटकरेंजिसके करेंजिसकेबाद जिसकेबादयह बादयहसभी यहसभीजातिया सभीजातियाअनुसूचित जातियाअनुसूचितजाति अनुसूचितजातिमें जातिमेंशामिल मेंशामिलहो शामिलहोगयी होगयीहैं। गयीहैं।बताते हैं।बतातेचलें बतातेचलेंकि चलेंकिउत्तर किउत्तरप्रदेश उत्तरप्रदेशके प्रदेशकेमुख्यमंत्री केमुख्यमंत्रीअखिले

Fig 5.5 Trigram file for Hindi text



इसे स्प्लिट करें
जिसके बाद यह सभी
 जातिया अनुसूचित जाति में शामिल हो गयी हैं।
 बताते चलें कि उत्तर प्रदेश के
 मुख्यमंत्री अखिलेश यादव ने कल के बाद आज भी लोक भवन में कैबिनेट की बैठक बुलाई थी।
जिसमें यह फैसला शामिल हैं।ज्ञात हो कि, बुधवार 21 दिसम्बर को भी मुख्यमंत्री अखिलेश
यादव ने कैबिनेट की बैठक बुलाई थी।
 जिसमें उत्तर प्रदेश सरकार की कैबिनेट ने कई महत्वपूर्ण प्रस्तावों
को मंजूरी दी गयी थी।

Fig 5.6 Unigram character file for Hindi text



इइससेे सस््पप््ललििटट ककररेेंं

जजििससककेे बबाादद ययहह ससभभीी

 जजााततििययाा अअननुुससूूचचिितत जजााततिि ममेेंं शशााममििलल हहोो गगययीी हहैैंं।।

 बबततााततेे चचललेेंं ककिि उउतत््ततरर पप््ररददेेशश ककेे

 ममुुखख््ययममंंतत््ररीी अअखखििललेेशश ययााददवव ननेे ककलल ककेे बबाादद आआजज भभीी ललोोकक भभववनन ममेेंं ककैैबबििननेेटट ककीी बबैैठठकक बबुुल

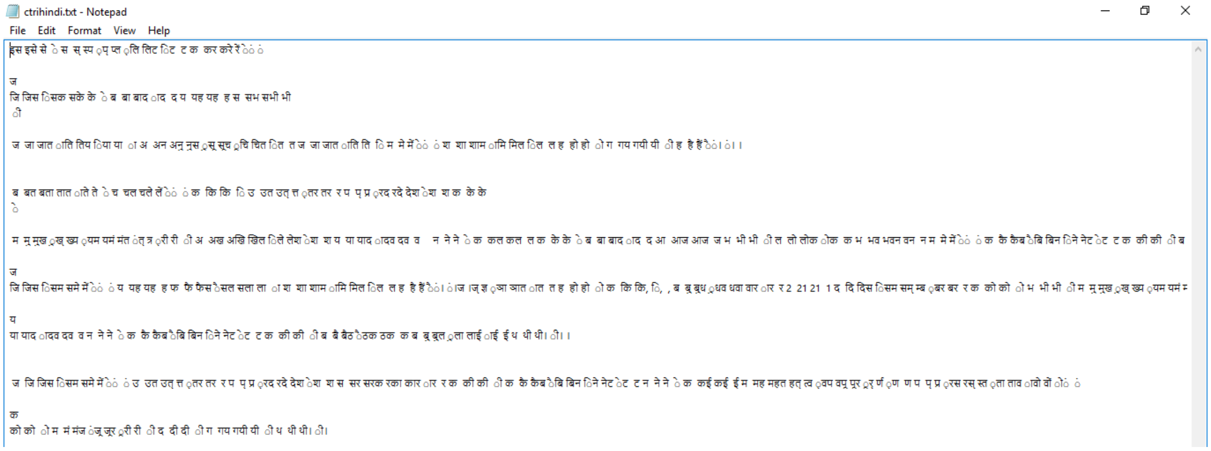
जजििससममेेंं ययहह फफैैससललाा शशााममििलल हहैैंं।।जज््ञञाातत हहोो ककिि,, बबुुधधववाारर 2211 ददििससमम््बबरर ककोो भभीी ममुुखख््ययममंंतत््ररीी अ

ययााददवव ननेे ककैैबबििननेेटट ककीी बबैैठठकक बबुुललााईई थथीी।।

 जजििससममेेंं उउतत््ततरर पप््ररददेेशश ससररककाारर ककीी ककैैबबििननेेटट ननेे ककईई ममहहतत््ववपपूूरर््णण पप््ररसस््ततााववोोंं

ककोो ममंंजजूूररीी ददीी गगययीी थथीी।।

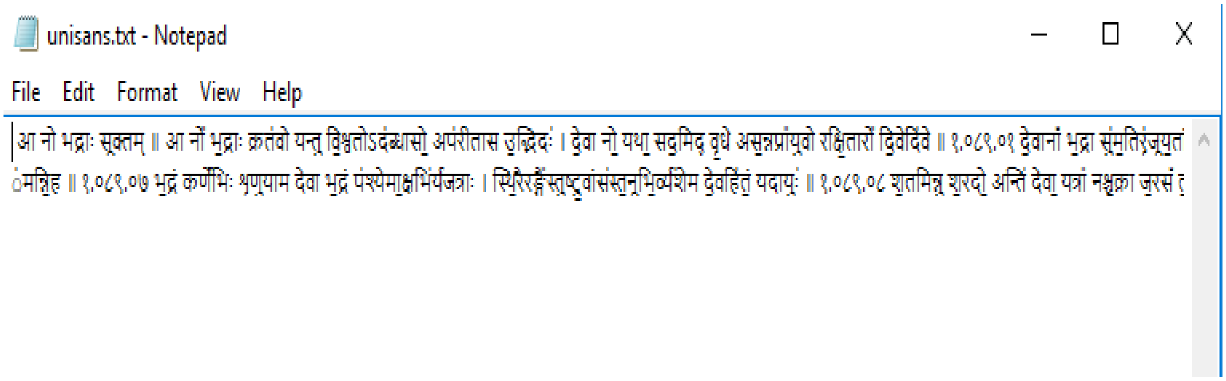
Fig 5.7Bigram character file for Hindi text



```
इस इसे से ेस स स् स्प ्प प्ल ्लि लिट िट ट क कर करे रें ें ं

ज
जि जिस िसक सके के े ब बा बाद ाद द य यह यह ह स सभ सभी भी
ी

 ज जा जात ाति तिय िया या ा अ अन अनु नुस ुसू सूच ूचि चित ित त ज जा जात ाति ति ि म मे में ें ं श शा शाम ामि मिल िल ल ह हो हो ो ग गय गयी यी ी ह है हैं ैं ं। ं। ।


 ब बत बता तात ाते ते े च चल चले लें ें ं क कि कि ि उ उत उत् त्त ्तर तर र प प् प्र ्रद रदे देश ेश श क के के
े

 म मु मुख ुख् ख्य ्यम यमं मंत ंत् त्र ्री री ी अ अख अखि खिल िले लेश ेश श य या याद ादव दव व  न ने ने े क कल कल ल क के के े ब बा बाद ाद द आ आज आज ज भ भी भी ी ल लो लोक ोक क भ भव भवन वन न म मे में ें ं क कै कैब ैबि बिन िने नेट ेट ट क की की ी ब

ज
जि जिस िसम समे में ें ं य यह यह ह फ फै फैस ैसल सला ला ा श शा शाम ामि मिल िल ल ह है हैं ैं ं। ं।ज ।ज् ज्ञ ्ञा ञात ात त ह हो हो ो क कि कि, ि, , ब बु बुध ुधव धवा वार ार र 2 21 21 1 द दि दिस िसम सम् म्ब ्बर बर र क को को ो भ भी भी ी म मु मुख ुख् ख्य ्यम यमं म

य
या याद ादव दव व न ने ने े क कै कैब ैबि बिन िने नेट ेट ट क की की ी ब बै बैठ ैठक ठक क ब बु बुल ुला ला लाई ाई ई थ थी थी। ी। ।


 ज जि जिस िसम समे में ें ं उ उत उत् त्त ्तर तर र प प् प्र ्रद रदे देश ेश श स सर सरक रका कार ार र क की की ी क कै कैब ैबि बिन िने नेट ेट ट न ने ने े क कई कई ई म मह महत हत् त्व ्वप वपू पूर ूर् र्ण ्ण ण प प् प्र ्रस रस् स्त ्ता ताव ावों वों ों ं

क
को को ो म मं मंज ंजू जूर ूरी री ी द दी दी ी ग गय गयी यी ी थ थी थी। ी।
```

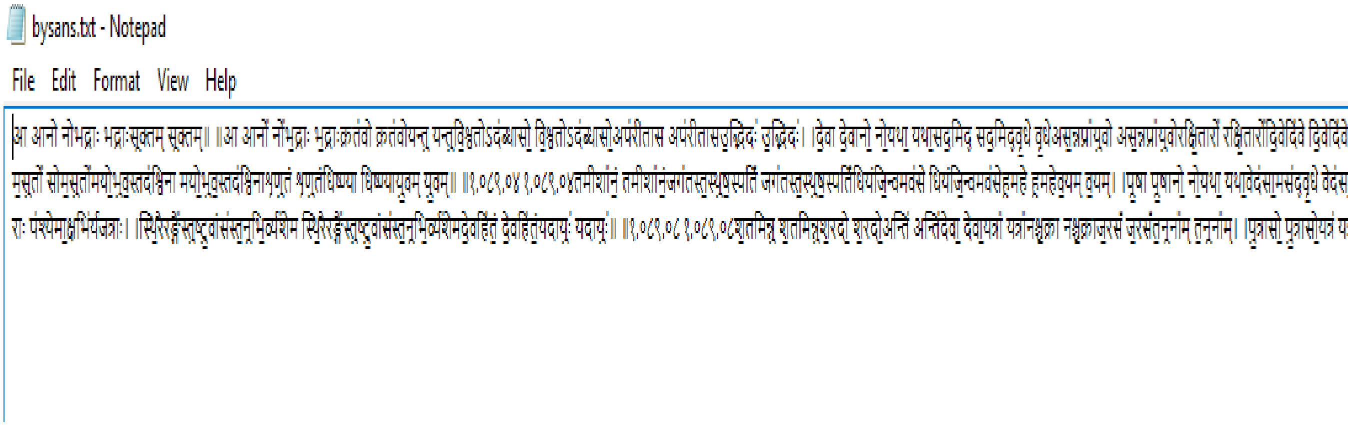
Fig 5.8 Trigram character file for Hindi text

Files related to Sanskrit text data:



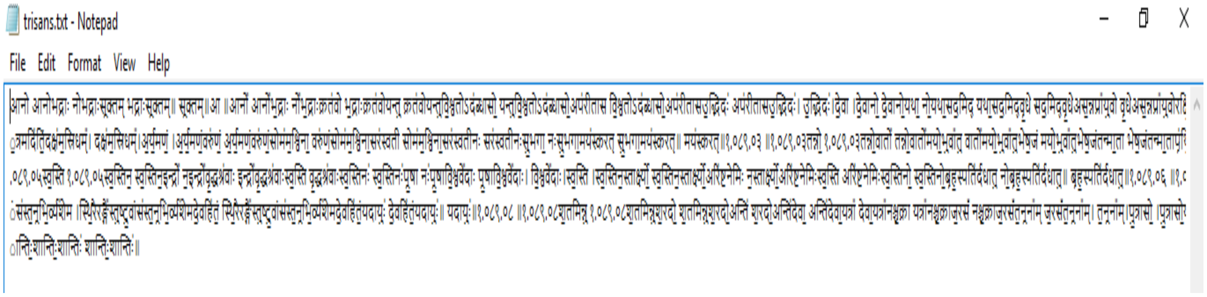
```
आ नो भद्राः सूक्तम् ॥ आ नो॑ भ॒द्राः क्रत॑वो यन्तु वि॒श्वतोऽद॑ब्धासो॒ अप॑रीतास उ॒द्भिदः॑ । दे॒वा नो॒ यथा॒ सद॒मिद्वृ॒धे अस॒न्नप्रा॑युवो रक्षि॒तारो॑ दि॒वेदि॑वे ॥ १.०८९.०१ दे॒वानां॑ भ॒द्रा सु॑म॒तिरृ॑जूय॒तां
◌मन्नि॒ह ॥ १.०८९.०७ भ॒द्रं कर्णे॑भिः शृणुयाम देवा भ॒द्रं प॑श्येमा॒क्षभि॒र्यज॑त्राः । स्थि॒रैरङ्गै॑स्तुष्टु॒वांस॑स्त॒नूभि॒र्व्य॑शेम दे॒वहि॑तं॒ यदायुः॑ ॥ १.०८९.०८ श॒तमिन्नु॒ श॒रदो॒ अन्ति॑ देवा॒ यत्रा॑ नश्च॒क्रा ज॒रसं॑ त॒
```

Fig 5.9 Unigram word file for Sanskrit text



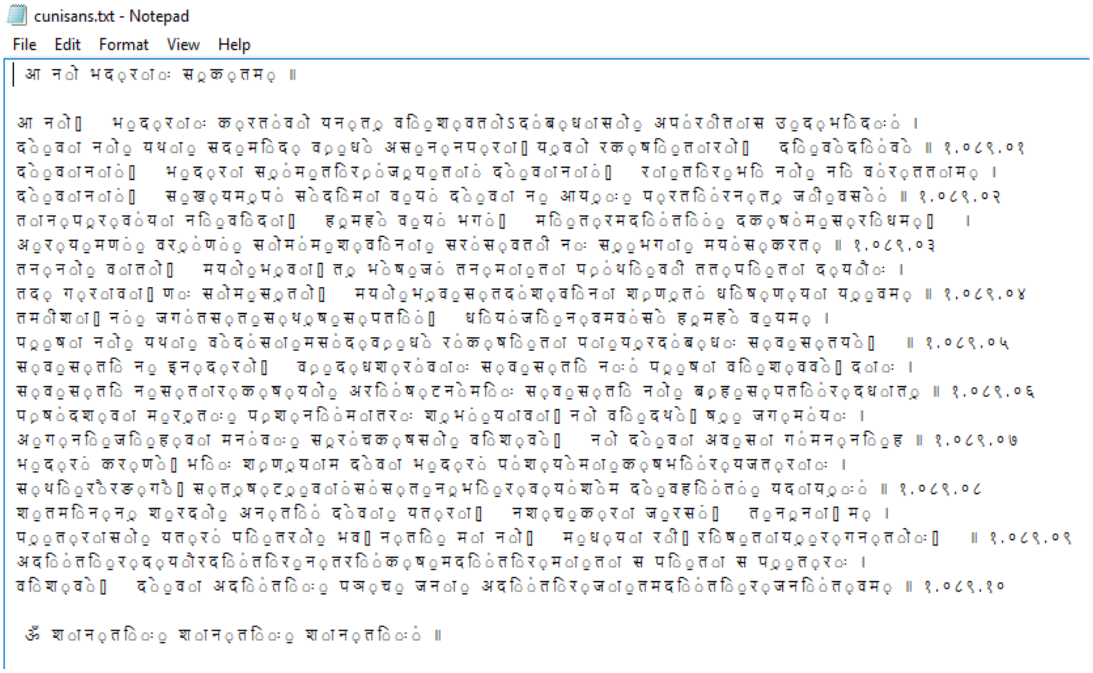
आ आनो नोभद्राः भद्राःसूक्तम् सूक्तम्॥ ॥आ आनो॑ नो॑भ॒द्राः भ॒द्राःक्रत॑वो क्रत॑वोयन्तु यन्तुवि॒श्वतोऽद॑ब्धासो वि॒श्वतोऽद॑ब्धासो॒अप॑रीतास अप॑रीतासउ॒द्भिदः॑ उ॒द्भिदः॑। ।दे॒वा दे॒वानो॒ नो॒यथा॒ यथा॒सद॒मिद् सद॒मिद्वृ॒धे वृ॒धेअस॒न्नप्रा॑युवो अस॒न्नप्रा॑युवोरक्षि॒तारो॑ रक्षि॒तारो॑दि॒वेदि॑वे दि॒वेदि॑वे
म॒सुतो॑ सोम॒सुतो॑मयो॒भुव॒स्तद॑श्विना मयो॒भुव॒स्तद॑श्विनाशृणुतं शृणुतंधिष्ण्या धिष्ण्यायु॒वम् यु॒वम्॥ ॥१,०८९,०४ १,०८९,०४तमीशा॑नं तमीशा॑नं॒जग॑तस्त॒स्थुष॒स्पतिं॑ जग॑तस्त॒स्थुष॒स्पतिं॑धियंजि॒न्वमव॑से धियंजि॒न्वमव॑सेहूमहे हूमहेव॒यम् व॒यम्। ।पू॒षा पू॒षानो॒ नो॒यथा॒ यथा॒वेद॑सा॒मस॑द्वृ॒धे वेद॑स
राः पश्ये॑मा॒क्षभि॒र्यज॑त्राः। ।स्थि॒रैरङ्गै॑स्तुष्टु॒वांस॑स्त॒नूभि॒र्व्यशे॑म स्थि॒रैरङ्गै॑स्तुष्टु॒वांस॑स्त॒नूभि॒र्व्यशे॑मदे॒वहि॑तं दे॒वहि॑तं॒यदायुः॑ यदायुः॑॥ ॥१,०८९,०८ १,०८९,०८श॒तमिन्नु श॒तमिन्नु॒शरदो॒ श॒रदो॒अन्ति॑ अन्ति॑देवा देवा॒यत्रा॑ यत्रा॑नश्च॒क्रा नश्च॒क्राज॒रसं॑ ज॒रसं॑त॒नूना॑म् त॒नूना॑म्। ।पु॒त्रासो॒ पु॒त्रासो॒यत्र॑

Fig 5.10 Bigram word file for Sanskrit text



आनो आनोभद्राः नोभद्राःसूक्तम् भद्राःसूक्तम्॥ सूक्तम्॥आ ॥आनो॑ आनो॑भ॒द्राः नो॑भ॒द्राःक्रत॑वो भ॒द्राःक्रत॑वोयन्तु क्रत॑वोयन्तुवि॒श्वतोऽद॑ब्धासो यन्तुवि॒श्वतोऽद॑ब्धासो॒अप॑रीतास वि॒श्वतोऽद॑ब्धासो॒अप॑रीतासउ॒द्भिदः॑ अप॑रीतासउ॒द्भिदः॑। उ॒द्भिदः॑।दे॒वा ।दे॒वानो॒ दे॒वानो॒यथा॒ नो॒यथा॒सद॒मिद् यथा॒सद॒मिद्वृ॒धे सद॒मिद्वृ॒धेअस॒न्नप्रा॑युवो वृ॒धेअस॒न्नप्रा॑युवोरक्षि
◌॒त्रमदि॑तिं॒दक्ष॑म॒स्रिध॑म्। दक्ष॑म॒स्रिध॑म्।अ॒र्य॒मणं॑ ।अ॒र्य॒मणं॒वरु॑णं अ॒र्य॒मणं॒वरु॑णं॒सोम॑म॒श्विना॑ वरु॑णं॒सोम॑म॒श्विना॒सर॑स्वती सोम॑म॒श्विना॒सर॑स्वतीनः सर॑स्वतीनः॒सु॒भगा॒ नः॒सु॒भगा॒मय॑स्करत् सु॒भगा॒मय॑स्करत्॥ मय॑स्करत्॥१,०८९,०३ ॥१,०८९,०३तन्नो॒ १,०८९,०३तन्नो॒वातो॑ तन्नो॒वातो॑मयो॒भुवा॑तु वातो॑मयो॒भुवा॑तुभे॒षजं॑ मयो॒भुवा॑तुभे॒षजं॑तन्मा॒ता भे॒षजं॑तन्मा॒तापृ॑
.०८९,०५स्व॒स्ति १,०८९,०५स्व॒स्तिन॒ स्व॒स्तिन॒इन्द्रो॑ न॒इन्द्रो॑वृ॒द्धश्र॑वाः इन्द्रो॑वृ॒द्धश्र॑वाःस्व॒स्ति वृ॒द्धश्र॑वाःस्व॒स्तिनः॑ स्व॒स्तिनः॑पू॒षा नः॑पू॒षावि॒श्ववे॑दाः पू॒षावि॒श्ववे॑दाः। वि॒श्ववे॑दाः।स्व॒स्ति ।स्व॒स्तिन॒स्तार्क्ष्यो॒ स्व॒स्तिन॒स्तार्क्ष्यो॒अरि॑ष्टनेमिः न॒स्तार्क्ष्यो॒अरि॑ष्टनेमिःस्व॒स्ति अरि॑ष्टनेमिःस्व॒स्तिनो॒ स्व॒स्तिनो॒बृह॒स्पति॑र्दधातु नो॒बृह॒स्पति॑र्दधातु॥ बृह॒स्पति॑र्दधातु॥१,०८९,०६ ॥१,०
◌॑संस्त॒नूभि॒र्व्यशे॑म ।स्थि॒रैरङ्गै॑स्तुष्टु॒वांस॑स्त॒नूभि॒र्व्यशे॑मदे॒वहि॑तं स्थि॒रैरङ्गै॑स्तुष्टु॒वांस॑स्त॒नूभि॒र्व्यशे॑मदे॒वहि॑तं॒यदायुः॑ दे॒वहि॑तं॒यदायुः॑॥ यदायुः॑॥१,०८९,०८ ॥१,०८९,०८श॒तमिन्नु १,०८९,०८श॒तमिन्नु॒शरदो॒ श॒तमिन्नु॒शरदो॒अन्ति॑ श॒रदो॒अन्ति॑देवा अन्ति॑देवा॒यत्रा॑ देवा॒यत्रा॑नश्च॒क्रा यत्रा॑नश्च॒क्राज॒रसं॑ नश्च॒क्राज॒रसं॑त॒नूना॑म् ज॒रसं॑त॒नूना॑म्। त॒नूना॑म्।पु॒त्रासो॒ ।पु॒त्रासो॒
◌न्तिः॒शान्तिः॒शान्तिः॑ शान्तिः॒शान्तिः॑॥

Fig 5.11 Trigram word file for Sanskrit Text

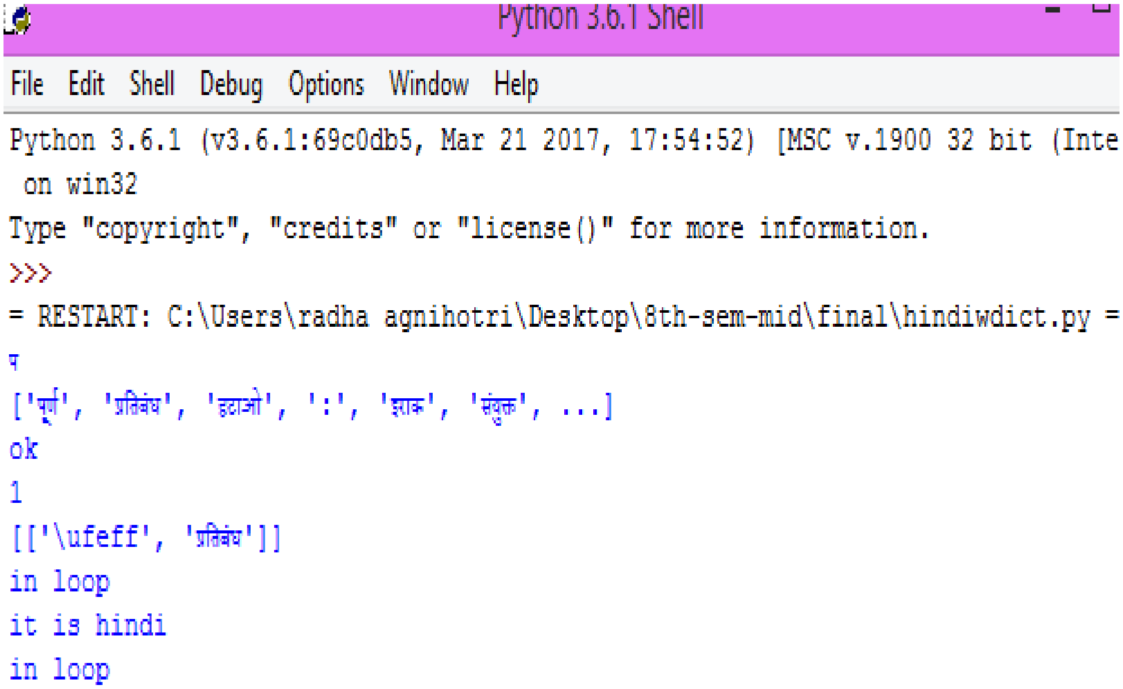


cunisans.txt - Notepad

```
आ नो भद्राः सूक्तम् ॥

आ नो॑ भ॒द्राः क्रत॑वो यन्तु वि॒श्वतोऽद॑ब्धासो॒ अप॑रीतास उ॒द्भिदः॑ ।
दे॒वा नो॒ यथा॒ सद॒मिद् वृ॒धे अस॒न्नप्रा॑युवो रक्षि॒तारो॑ दि॒वेदि॑वे ॥ १.०८९.०१
दे॒वानां॑ भ॒द्रा सु॑म॒तिरृ॑जूय॒तां दे॒वानां॑ रा॒तिर॒भि नो॒ नि व॑र्तताम् ।
दे॒वानां॑ स॒ख्यमुप॑ सेदिमा व॒यं दे॒वा न॒ आयुः॒ प्रति॑रन्तु जी॒वसे॑ ॥ १.०८९.०२
तान्पूर्व॑या नि॒विदा॑ हूमहे व॒यं भगं॑ मि॒त्रमदि॑तिं॒ दक्ष॑म॒स्रिध॑म्॑ ।
अ॒र्य॒मणं॒ वरु॑णं॒ सोम॑म॒श्विना॒ सर॑स्वती नः सु॒भगा॒ मय॑स्करत् ॥ १.०८९.०३
तन्नो॒ वातो॑ मयो॒भु वा॑तु भेष॒जं तन्मा॒ता पृ॑थि॒वी तत्पि॒ता द्यौः ।
तद्ग्रावा॑णः सोम॒सुतो॑ मयो॒भुव॒स्तद॑श्विना शृणुतं धिष्ण्या यु॒वम् ॥ १.०८९.०४
तमीशा॑नं॒ जग॑तस्त॒स्थुष॒स्पतिं॑ धियंजि॒न्वमव॑से हूमहे व॒यम् ।
पू॒षा नो॒ यथा॒ वेद॑सा॒मस॑द्वृ॒धे र॑क्षि॒ता पा॒युरद॑ब्धः स्व॒स्तये॑ ॥ १.०८९.०५
स्व॒स्ति न॒ इन्द्रो॑ वृ॒द्धश्र॑वाः स्व॒स्ति नः॑ पू॒षा वि॒श्ववे॑दाः ।
स्व॒स्ति न॒स्तार्क्ष्यो॒ अरि॑ष्टनेमिः स्व॒स्ति नो॒ बृह॒स्पति॑र्दधातु ॥ १.०८९.०६
पृष॑दश्वा म॒रुतः॒ पृश्नि॑मातरः शु॒भंया॒वानो॑ वि॒दथे॑षु॒ जग्म॑यः ।
अ॒ग्नि॒जि॒ह्वा मन॑वः॒ सूर॑चक्षसो॒ विश्वे॑ नो दे॒वा अव॒सा ग॑मन्नि॒ह ॥ १.०८९.०७
भ॒द्रं कर्णे॑भिः शृणुयाम देवा भ॒द्रं प॑श्येमा॒क्षभि॒र्यज॑त्राः ।
स्थि॒रैरङ्गै॑स्तुष्टु॒वांस॑स्त॒नूभि॒र्व्य॑शेम दे॒वहि॑तं॒ यदायुः॑ ॥ १.०८९.०८
श॒तमिन्नु श॒रदो॒ अन्ति॑ देवा॒ यत्रा॑ नश्च॒क्रा ज॒रसं॑ त॒नूना॑म् ।
पु॒त्रासो॒ यत्र॑ पि॒तरो॒ भव॑न्ति॒ मा नो॑ म॒ध्या री॑रिषता॒युर्गन्तोः॑ ॥ १.०८९.०९
अदि॑ति॒र्द्यौरदि॑तिर॒न्तरि॑क्ष॒मदि॑तिर्मा॒ता स पि॒ता स पु॒त्रः ।
विश्वे॑ दे॒वा अदि॑तिः॒ पञ्च॒ जना॒ अदि॑तिर्जा॒तमदि॑ति॒र्जनि॑त्वम् ॥ १.०८९.१०

ॐ शान्तिः॒ शान्तिः॒ शान्तिः॑ ॥
```

Fig 5.12 Character file for unigram Sanskrit

Output after running the code on the python shell



Python 3.6.1 Shell

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Inte
 on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:\Users\radha agnihotri\Desktop\8th-sem-mid\final\hindiwdict.py =
प
['पूर्ण', 'प्रतिबंध', 'हटाओ', ':', 'झाड़', 'संयुक्त', ...]
ok
1
[['\ufeff', 'प्रतिबंध']]
in loop
it is hindi
in loop
```

Fig 5.13 Python shell to identify the language

REFERENCES

[1] Mallamma V Reddy,” Natural Language Identification and Translation Tool for Natural Language Processing.” International Journal of Science and Applied Information Technology, ISSN No. 2278-3083, Vol. 1, No.4, September October 2012.

[2] Deepamala N, Ramakanth Kumar P, ” Language Identification of Kannada Language using N-Gram. ” IJCA , Vol. 46 , No. 4, May 2012.

[3] Yew Choong Chew, Yoshiki Mikami, Robin Lee, ” Language Identification of Web Pages Based on Improved N-gram Algorithm, ” in IJCSI International Journal of Computer Science , Vol. 8, Issue 3, No. 1, May 2011

[4] Vatanen, Tommi and Vayrynen, Jaakko J and Virpioja, Sami, ”Language Identification of Short Text Segments with N-gram Models.” European Language Resources Association, 2010.

[5] Mallikarjun Hangarge , B.V. Dhandra, ”Offline Handwritten Script Identification in Document Images.” International Journal of Computer applications (0975 8887) Vol. 4 No.6, July 2010

[6] Steven Bird, Ewan Klein, and Edward Loper, ” Natural Language Processing with Python-Analyzing Text with the Natural Language Toolkit. ” O’Reilly Media, 2009

[7] M C Padma, Dr P A Vijaya,” Language identification of kannada, Hindi and English text words through visual discriminating features.” in international journal of computational intelligence systems, vol.1, No. 2, may, 2008, pp. 116 – 126

[8] Magnus Lie Hetland , ” Beginning Python: from novice to professional ”, 2008

[9] Daniel Jurafsky, James H. Martin , ” Speech and Language Processing:An introduction to natural language processing, computational linguistics, and speech recognition.” Prentice Hall,Englewood Cliffs, New Jersey,2006.

[10] Kavi Narayana Murthy and G. Bharadwaja Kumar, ” Language Identication from Small Text Samples. ” Journal of Quantitative Linguistics Vol. 13, No. 1, 2006, pp. 57 – 80

[11] B. Ahmed, S. Cha, ”Language Identification from Text Using N-gram Based Cumulative Frequency Addition”, Proceedings of CSIS 2004, Pace University, May 7th, 2004

[12] Gregory Grefenstette, ”Comparing two language identification schemes.” JADT 1995, 3rd Internationl conference on statistical analysis of textual data, Rome, Dec 11-13,1995.

[13] Ted Dunning, ”Statistical identification of language.” Technical Report MCCS94273, Computing Research Lab, New Mexico State University,

1994

[14] Simon Kranig, ”Evaluation of Language, Identification Methods.”University of T’ubingen, International Studies in Computational Linguistics.

[15] Rick Briggs, ”NASA article on Sanskrit in AI.” Spring , 1985.

[16] http://www.differencebetween.com/difference-between-indianlanguages-sanskrit-and-vs-hindi.

[17] http://hi.wikipedia.org (Hindi Wikipedia)

[18] http://sa.wikipedia.org (Sanskrit Wikipedia)