Transliteration Verification of English-Hindi Words using Machine Learning Approach

MSc Research Project
Data Analytics

Mayank Jain
x17104769

School of Computing
National College of Ireland

Supervisor: Dr Catherine Mulwa

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Mayank Jain
Student ID:	X17104769
Programme:	Data Analytics
Year:	2018
Module:	MSc Research Project
Lecturer:	Dr Catherine Mulwa
Submission Due Date:	13/08/2018
Project Title:	Technical Report
Word Count:	6728

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	15th September 2018

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Transliteration Verification of English-Hindi Words using Machine Learning Approach

Mayank Jain
x17104769
MSc Research Project in Data Analytics

15th September 2018

Abstract

Automatic transliteration and back-transliteration across languages with different phonemes and alphabets inventories such as English-Hindi, English-Chinese, and English-Korean and few more have practical importance in cross lingual information retrieval, machine translation and automatic bilingual dictionary compilation. Proper name pronunciation based translation termed as transliteration, for many multilingual natural language processing task such as cross lingual information retrieval and statistical machine translation is very important in this modernised world. Due to the pronunciation difference between the source and target language, this task is very challenging. In past for transliteration, research efforts has demonstrated a 30-40 percent error using the top-1 reference. For many applications, this error leads to performance degradation. In asian countries especially in India, there has been very less work which has been done in the field of English-Hindi transliteration. The motive of this research, is to develop a novel and an efficient model , to verify a given proper name transliteration using different machine learning approach.

1 Introduction

Machine translation, an essential component and highly demanded technology which is essential in many multilingual applications. The main application in todays global environment requires the cross lingual question answering and cross lingual information retrieval. In any language, proper names forms an open set and they are shown to grow the size of the corpora. For many multilingual natural language processing tasks, multilingual spoken document retrieval and cross lingual information retrieval (CLIR), proper name transliteration is very important. Conventionally dictionaries which are an aid to the human translation are used for dictionaries based machine translation and about 50,000 to 150,000 entries are been contained by the typical dictionaries. For example, a collection of text from the press newswire has 50 million words comprising 200000 distinct English words. The Out of vocabulary (OOV) words are typically names, like places, and products names. Thus in such cases where the OOV are spelled out, Transliteration is required. Transliteration with approximate equivalence of pronunciations into target language is always used to translate source names. On the basis of the single syllable, current DOM, which is termed as Direct Orthographical Mapping approach does alignment and segmentation. Lots of eyeball in NLP processing such as Information extraction, Information retrieval and Machine translation gets attracted by the name entities. For personal and place names, source names are always considered into foreign language with the similar pronunciations. Thus for a given name in source language , machine transliteration task is translating it into target language with (i) conforms to the phonology of the target language , (ii) phonemically close to the source name and (iii) in the target language, matches the user intuition of the equivalent of the source language name. Many languages, but particularly Indian languages using the non-roman scripts are inscribed. That is the reason there is a huge demand of development of an effective and efficient transliteration for such languages.

Forward transliteration, a process of transliterating a native language to its foreign counter parts whereas transliterating the words back in the native counterpart is termed as backward transliteration. However, the dearth of standard set of rules leads to variations of spelling in the target language and thus creates a problem. Also, due to multilingual datas rapid growth proper name as grew in abundance and in the lexicon they had become indistinct. The machine learning topic for several different language pairs has been studied extensively. Based on nature of language, various methodologies have been developed for machine based transliteration. Most of the current transliteration systems, based on the alignment for transliteration uses the generative model. Around a decade ago to deal with the proper name and technical terms, machine transliteration emerged as a part of machine translation. Its a sub field of computational linguistics and the nature of tasks language gets specific by the language processing of transliteration methods. More and more proper names got appeared and becomes undefined lexicon because of the fast growth of multilingual information. Thus these undefined proper names are termed as OOV words, which is known as out of vocabulary and the performance of Machine transliteration (MT) and CLIR gets adversely effected by these words. If the proper names are better handled then only the performance of CLIR can be improved. On the nature of languages considered for machine transliteration various methodologies have been developed. Machine transliteration, in multi lingual world is an increasingly important problem as well as it plays a crucial role in many downstream applications like cross lingual information retrieval system and machine translation (Das et al.; 2009).

According to the researchers, use of different phonemes and alphabets in the transliteration is very challenging sometimes. Phoneme: Its a speechs smallest unit which distinguishes the meanings. They are very important and substituting them causes meaning of word to change. For example the transliteration between English and Arabic or English and Chinese is not at all trivial. The huge difference between the source and target language pronunciation doesnt have a corresponding phoneme. The phonetic structure should be preserved as closely as possible and thus the designing of the machine transliteration. These Source and target languages pronunciation leads to one to many, many to one or sometimes one to one phoneme mappings. Thus the researchers have found the best automatic ways for transliteration. Transliteration current models can be classified as phoneme based models and grapheme based model. Using 10 references best performance will be approx. 70 percent alphabet edit distance error. The error rate can be high in case of usage of one reference. Not only this, the error rate will be higher if the whole proper name is being used one unit and only one reference is getting used. Thus, the transliteration verification is very much required as it is drawing limited attention to many applications and especially in Asian countries.

1.1 Research Question

RQ- How can we improve (accuracy) transliteration of Hindi-English words using machine learning model to support Indian people who novice in understanding the English language

To address the research question, the following objectives are specified and tackled.

1.2 Research Objectives

Objective 1 is state of the art review of transliteration of Hindi to English Words from 2002 to 2018.

The research objective (Objective 2): specification, design, implementation and results of Decision tree model machine learning model used during transliteration of Hindi-English words.

The third objective (Objective 3) is: Implementation, evaluation and results of Random Forest model Hindi-English words.

The Fourth objective (Objective4) is: A comparison of develop Decision Tree machine model (Objective2) with developed Random Forest (Objective 3).

2 Literature Review

In recent years machine transliteration in Asia has received a drastic research attention. In most of the cases, the source and the target languages have been English and Indian languages respectively. Generally machine transliteration takes place in two different direction, Backward and Forward. If A is a word taken from one language and B is taken from another language, then conversion of A to B is forward transliteration while conversion of B to A is backward transliteration. This section will provide the brief description about the important and main terminology which has been used and also about the studies made as part of this project.

According to (Oh et al.; 2006a) , there are four different types of transliteration models which are Phoneme based transliteration model, Grapheme based transliteration model, Correspondence based transliteration model and Hybrid based transliteration model. All these models uses the different units for transliteration. For example, the unit which is used in phoneme based is pronunciation units i.e. phoneme, which uses 2 steps. The 1st step is the conversion of grapheme to phonemes and then in 2nd step the conversion of phonemes are again converted to grapheme. Similarly the unit which is used in grapheme model is word segments, i.e. Grapheme

2.1 A Review of Grapheme Based Model

Grapheme- In written languages, its a fundamental unit which includes numerals, alphabetical letters, characters and all the individual symbols of any writing systems. A grapheme respond to 1 phoneme, especially in phonemic orthography. Example like word ship contains 4 graphemes (s, h, i, and p) but the phoneme will be only three because sh is termed as a digraph. Diagraphs are the representation of multiple spellings, like /sh/, /tr/ etc

The grapheme based model ψ G, from source graphemes to target graphemes is direct orthographical mapping. Based on ψ G, many transliteration model have been proposed like based on decision trees, source channel model, a Joint source channel model and Transliteration model. In the source channel model proposed by (Haizhou et al.; 2004), deals with English- Korean transliteration. The chunk of graphemes were used which corresponded to source phoneme. First the Chunk of English graphemes were segmented by the English words and after that all possible chunks of Korean graphemes which were corresponding to the English graphemes chunks were produced. And at last Korean graphemes most relevant sequence was identified by using the source channel model. The main advantage was that the chunk of graphemes representing source languages phonetic property was considered in this approach. Similarly (Kang and Kim; 2000) proposed the methods which were based on English to Japanese transliteration and in their model chunk of phonemes are represented as a node and the graphemes are represented as graphemes. Just like the methods which was based on source channel model the phonetic aspect was considered in the form of chunks of graphemes.

2.2 Phoneme based transliteration model

Phoneme. - It is the smallest unit of speech in which the meanings gets distinguish. (Teshome et al.; 2015) showcased that the transliteration quality should always be high. They had considered the application of statistical method to automatic Machine Translation from English to Amharic in their research and focused on improving the translation quality by applying phonetic transcription on the target side. These units are very important and the words meaning can get change if phoneme gets substituted. For example, if the sound of the letter [p] and [b] gets substituted in the word pig, then it will transform to word big. Therefore the letter /p/ is the phoneme. Also, the sounds smallest segment is known as phone, the physical realization of phoneme. A study of sound of humans speech which is concerned with the actual properties of the audition, production and perceptron of speech sound is termed as Phonetics.(**?**) Phonetics independently deals with the sound rather than the context. While on the other hand phonology studies the sound systems such as distinctive features, phonemics and phonological rules. Therefore phonology is language specific while the definition of phonetics applies across languages.

In the ψ P, as mentioned above, the transliteration key which is used in the phoneme based transliteration model is pronunciation or the source phoneme. There were many work done by the researchers in the domain to phoneme based transliteration and several methodologies have been implemented. By combining several parameters with weighted finite state transducers (Knight and Graehl; 1998) modelled the Japanese to English transliteration. In fact (Stalls and Knight; 1998) developed the same model for Arabic to English transliteration. Among those there was a special model which was developed by (Oh et al.; 2006b) by the extended Markov window model. In his model, by using the pronunciation dictionary, English words were transformed to English pronunciations and then the English phonemes were segmented into the chunks of English phonemes and each chunk as defined by handcrafted rule which was corresponded to Korean grapheme.

2.3 Hybrid and Correspondence-based transliteration Models

In machine transliteration, the attempts to use both source phoneme and source graphemes led into the hybrid transliteration model ψ H and correspondence based transliteration model ψ C. ‘The use of correspondence is made between the source phoneme and source grapheme when it produces the target graphemes, the latter through linear interpolation ψ P and ψ G were combined (Oh et al.; 2006c) . Many researchers proposed the hybrid based transliteration with the weighted finite state transducers model and through the linear interpolation they combined the ψ G and ψ P. In their ψ P, several parameters such as source grapheme to source phoneme probability, target language probability and source phoneme to target grapheme, probability were considered. And in the ψ G , the probability of source grapheme to target grapheme was mainly considered. The dependence between the source phoneme and source grapheme was not taken into consideration in the combining process and which was the biggest disadvantage of the hybrid model. Thus model proposed by (Oh et al.; 2006c), by using the correspondence between source phoneme and source grapheme considered the mentioned dependence. Also among some of the important terms used in Machine based transliteration includes Cross Lingual Information retrieval, which is basically an information retrieval subfield which deals with the information written in a language which is different from the users query language. Cross lingual information retrieval has many other synonyms like Trans lingual information retrieval, cross language information retrieval, multilingual information retrieval.

Among the different writing system, The Syllabary system here is defined as a set of written symbols which represents syllables that constitute words. Consonant sound which is followed by the vowel sounds are represented for symbols in syllabary. Similarly in feature writing system, symbols which are contained by FWS, do not represent whole phoneme, but rather the features which constitute phoneme. In alphabetic or segmental writing systems, a small set of alphabet or letter is being represented by phoneme of spoken languages. For example, the Latin and Arabic writing system are segmental.

According to (Sunitha and Jaya; 2015), the simplest method is the grapheme model because the graphemes are mapped directly to the destination languages and thus it is known as direct method because it doesnt uses any additional knowledge. Both correspondence based model and hybrid based model uses phoneme and grapheme from source language transliteration. But according to (Oh et al.; 2006c) these two methods are more complicated to implement compared to the others.

There are many works and researches which has been done on machine transliteration and the problem which has been often viewed was translation problem. has mentioned in their research about the comparison and testing of dierent Machine Transliteration Models. They tested the 4 main types of models which are phoneme, grapheme, correspondence and hybrid for English to Korean and English to Japanese transliteration and found that the performance of the machine transliteration can be improved by two ways:

1) Produce a list of transliteration by combining the multiple transliteration models.

2) On the basis of the relevance, rank the transliterations.

While (Wei and Bo; 2008) proposed a novel method for Chinese to English transliteration by using a weighted finite state transducers (WFST). In their work they have built a phoneme based, grapheme based and an extended phoneme based model and combined these models with unified framework of weighted finite state transducers for Chinese English transliteration.

2.4 Transliteration in India

In spite being one of the richest areas in terms of linguistic diversity India has lot in common. Our model will be based on English-Hindi proper name transliteration. Major Indian languages use the scripts which are developed from Brahmin or Indic script family. According to (Mathur and Saxena; 2014) for Indian languages, machine translation is still in its infancy. They have specifically used English-Hindi language pair for and applied hybrid approach in which extraction of individual phonemes from the words using the rule based method has been achieved and then by applying the statistical approach the English phoneme gets converted to Hindi phoneme yielding in very less accuracy. Similarly, (Surana and Singh; 2008) proposed a more discerning method in which they have not used the training data on the target side instead they used the fuzzy strings matching and more sophisticated technique there. Their model was based on the transliteration of Foreign and Indian languages to the Hindi and Telugu and have achieved .44 in MRR.

As India has more than 72 inside languages, thus there is a huge scope of transliteration. Dr Soman KP in his literature has addressed developments in Indian Language Machine transliteration as according to him, for many Natural language processing (NLP) applications, they are considered as an important task (Antony and Soman; 2011). English to Kannada transliteration, English to Malayalam transliteration, English to Telugu transliteration and English to Hindi transliteration were the main focus of his survey and in which the different techniques and models which were used were deeply explained. As mentioned above that there are several languages in India thus the need of transliteration is at peak, the same thing has been mentioned by (Emeneau; 1956) in his paper that among Indian languages there are lot of similarities, even though they belong to different families. From the linguistic point of view, there has been lot of work done on the writing systems. The below figure 1 is the phonetically arranged basic consonants in the alphabet.

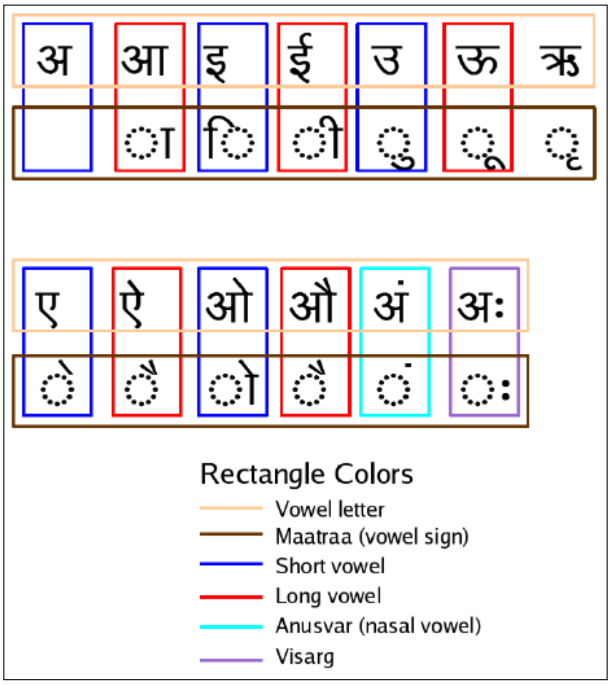


Figure 1: Phonetically arranged vowels

The most important work in the category of linguistics is on encoding and alphabets for Indian Languages. The development of a standard For Brahmin origin scripts which was the most important work in that category is Indian Standard Code for Information Interchange (ISCII). For South Asian languages, this encoding which is also termed as super encoding was not successful because of the unsuccessful adoption for computing.

But in spite of its failure fact, it was an important work because of the similarities among the Brahmin origin scripts alphabets. One of the other important work was the encoding standard UTF-8, which was based on the Unicode (Pakray and Bhaskar; 2013). The UTF-8 is quite close to ISCII standard except the fact of different slots for different scripts. There were limitations which were pointed out in both UTF-8 and ISCII with respect to the Indian languages. Still from the researchers point of view, both of these encodings have very strong advantages over any other encoding for Indian Languages. Computational analysis of Brahmin scripts by (Sproat; 2003) was mainly towards the shape of letters i.e. grapheme. (Singh; 2006) in his model of phonology-orthography interface, argued that Akshar is the minimal articulatory unit of speech. The main focus of his work in Hindi orthography was to suggest proposals to reform the orthography. (Rama et al.; 2009) in his paper has mentioned about the great deal of debate about the issues like, whether Phoneme are real or not. For example phonemes come from letters, argued by one philosopher. The fact seems to be tilting towards the view about phonemes are useful fictions and the influence of orthography was the stem for their psychological reality.

2.5 Phonetic Models Need for Brahmin Script

There are many important reasons that why an abstracted model of Brahmin Origin scripts was needed, some of them are:

- Compared to European languages, there is much less standardization for these languages, which means that there can be more than one spelling for the same word. For example, in dictionary, it is not practically possible to list all the variants somewhere. In fact about valid and non-valid spellings there may not be an agreement
- For all the languages which uses Brahmin scripts, it is better to have one model and one text processing model.
- In these languages, there are dialectal variants of words and each languages has many dialects. The writer when writing the standard language, may use a spelling influenced by her first language or dialect.
- One of the reason of phonetic models need was the low literacy rate and for many practical purposes use of English, as Indians including highly educated ones may not have much writing practice.
- In Indian languages, there are large number of Cognate words, like words borrowed from Persian, Sanskrit words, words from English etc. Thus there should be some mechanism to identify these words automatically.

This model focuses on the verification of proper name transliteration. One wants to verify in the given name pair, that whether the pair refers to the same proper name or not which is one from the source language and other from the target language. To set the rejection threshold to an optimal level, the similarity score should be robust and reliable. The main objective of this model is to explore approaches with low complexity and high accuracy. English and Hindi proper names have been chosen in this research. As there is very less researches which has been done in India for transliteration of English-Hindi by using Decision trees and Random forest. Thus this projects researches about these two models and predict whether they will be efficient for the transliteration purpose.

A proper name based on pronunciation gets translated to the proper name transliteration intuitively. An extreme good evaluation method should be provided from the phonetic based edit distance method. There might be different base phone set in the source and the target language. Thus for edit distance measurement, one has to convert these phone set to a unified phone set. There is a requirement of similarity measure for each phoneme pair instead of treating all errors with unique cost. Also there should be a data driven approach instead of manually identification the phone mapping rules. For that models like Decision Trees and Random Forest Machine learning model will come into extinct.

3 Methodology

This section presents the techniques and scientific methods which is used in this research. A transliteration process flow diagram has been designed including technical design of the implementation and evaluation of the model is presented in this section. The data mining method which is used here is CRISP-DM and used to guide the candidate during the implementation.

3.1 Data Mining Methodology

CRISP-DM which is termed as cross industry process for data mining methodology is used in this research. As it is a robust and well-proven technology and so for easy revision and well-structured document, the CRISP-DM has been planned in this project. CRISP-DM consists of six stages which are as follows:

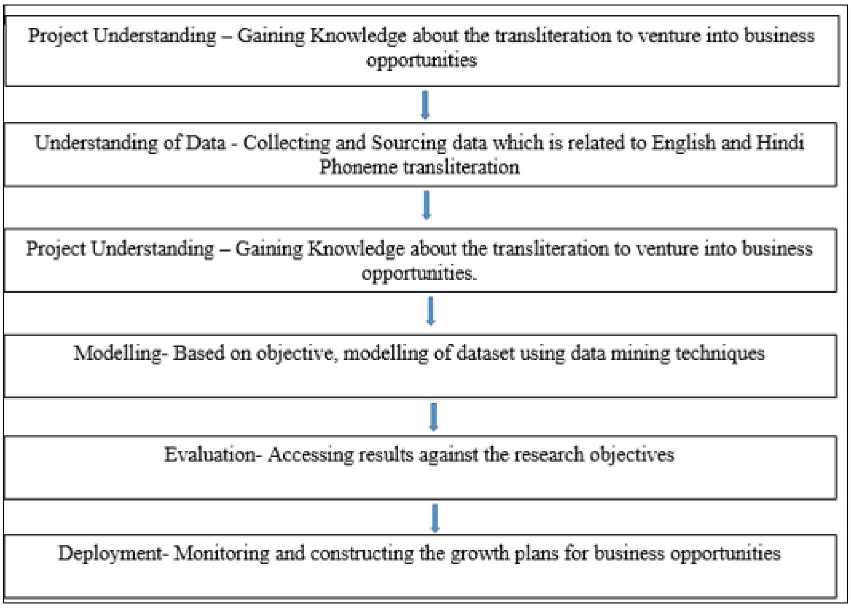

Project Understanding – Gaining Knowledge about the transliteration to venture into business opportunities
Understanding of Data - Collecting and Sourcing data which is related to English and Hindi Phoneme transliteration
Project Understanding – Gaining Knowledge about the transliteration to venture into business opportunities.
Modelling- Based on objective, modelling of dataset using data mining techniques
Evaluation- Accessing results against the research objectives
Deployment- Monitoring and constructing the growth plans for business opportunities

Figure 2: CRISP-DM Methodology

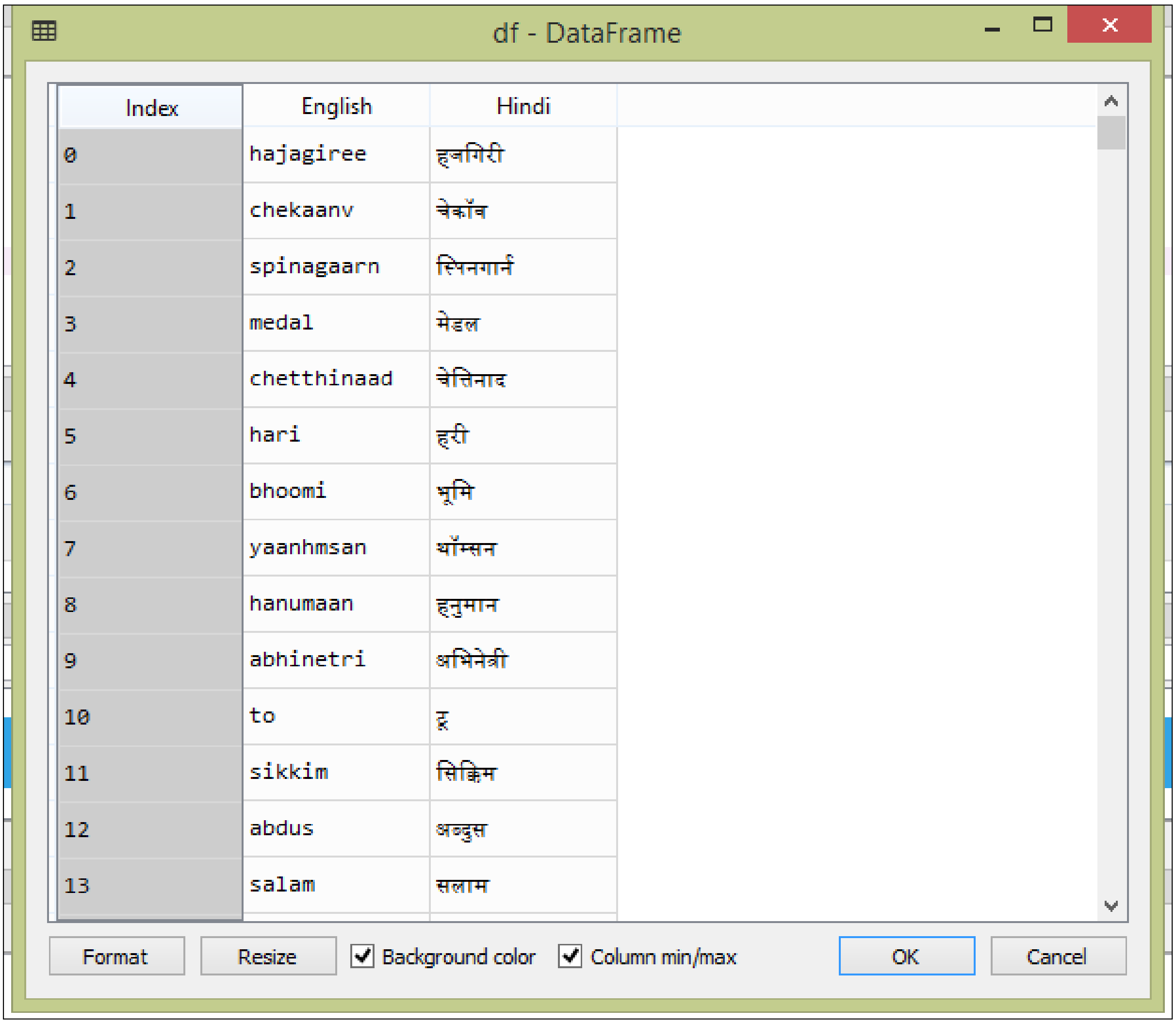
As per the figure 2 above, the very first step includes with the understanding of the project and gaining the knowledge about the transliteration. Understanding of the data and its attributes make the most important valuable form of CRISP-DM. Similarly the other steps includes project understanding and gaining the knowledge about the transliteration so that it can venture into business opportunities. Modelling, Evaluation and Deployment are the other major technical steps which strengthen the model according to the desired output.

3.2 Data Pre-Processing

As in this research, English and Hindi proper name transliteration has been opted so to conduct the experiment, a parallel corpus of English and Hindi proper names pair dataset was downloaded from Github.com and the link of the dataset is provided in the reference section. The data was in the text format and thus the data was first converted to excel and encoding was set to UTF-8. The data which was downloaded from the website was the raw data and contains many missing and NA values. Not only this the data was consisting of many outliers and also some symbols like brackets, comma, numbers. Thus by using the python programming in spyder version (anaconda), these unwanted elements has been removed. After that, to make the dataset ready for the experiment, among those words pairs, approximately 14879 pairs were used for the training and about 3719 pairs were used in testing.

3.3 Data Transformation

The data which has been taken from github was a raw data and is in the form of text file. The next step includes a major step where the encoding was set to UTF-8. As the data is about the transliteration and consists of English and Hindi words pair thus first step was converting the text file into xlsx format and implying the encoding UTF-8, so that it can be in a machine readable form. The below given figure 3 explains about the overview of the data after encoding where the first column consists of the English words and the second column consists of the Hindi form of that English words



Index	English	Hindi
0	hajagiree	हजगिरी
1	chekaanv	चेकॉंव
2	spinagaarn	स्पिनगार्न
3	medal	मेडल
4	chetthinaad	चेत्तिनाद
5	hari	हरी
6	bhoomi	भूमि
7	yaanhmsan	थॉम्सन
8	hanumaan	हनुमान
9	abhinetri	अभिनेत्री
10	to	टू
11	sikkim	सिक्किम
12	abdus	अब्दुस
13	salam	सलाम

Figure 3: Dataset

The major step which has taken here regarding the encoding was to import the label encoder package inn python. As the data consists of Hindi words which has a different encoding parameter. While performing the pre-processing, machine was giving an error which was about the English-Hindi words. Thus by the help of label encoding the non-numerical data gets converted into the numerical form so that it can be in the machine readable form.

3.4 Analysis Tool

The experiment results and graphs were produced in the open source programming language Python version 3.6. The tools and packages which were installed during the implementation were:

Anaconda- An open source distribution of python programming language for data science and machine learning related application.

Spyder- For scientific programming in Python language, spyder is an open source cross-platform integrated development environment. Spyder integrates with a number of prominent packages in the python stack like NumPy, Matplotlib, SciPy , etc.

Rapid Miner- As rapid miner provides the data mining and machine learning procedure thus to get an overview and idea rapid miner was installed

3.5 Project Process Flow Diagram

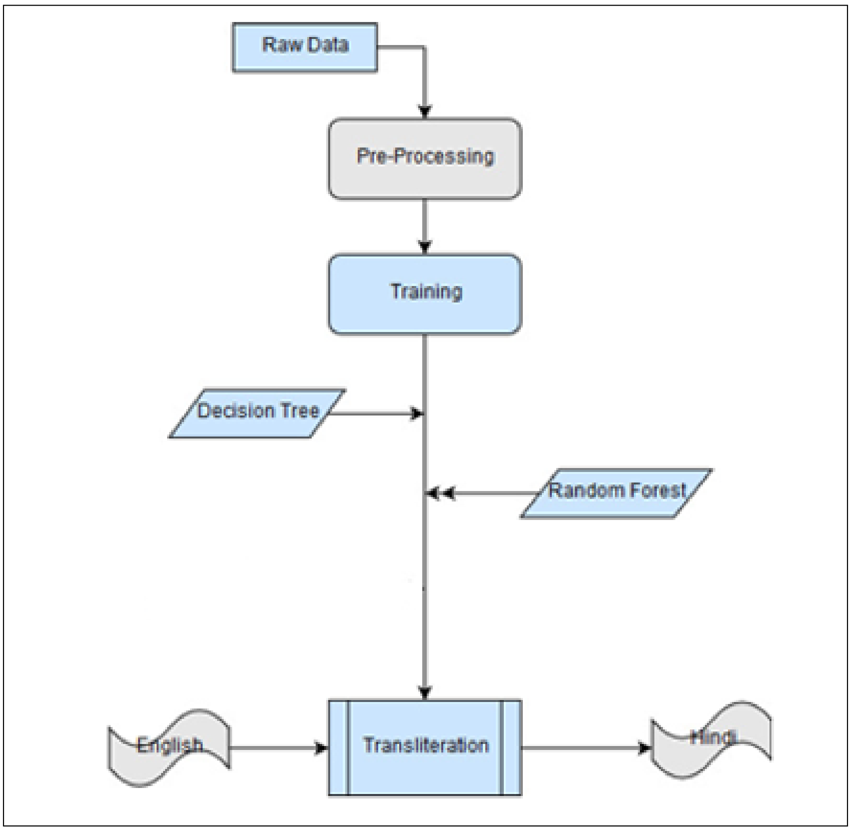


Figure 4: Process Flow Diagram

4 Implementation

4.1 Introduction

Our goal is to find the best machine learning model for the transliteration of English-Hindi name pairs and checking the words alignment accuracy by comparing it by different models. This process of implementation includes several phases which are gathering the data, development of proposed data mining models, data analysis, analysis and comparison of results with the other machine learning models. The complete description about each phase is described and explained in the following sections:

4.2 Training and Testing of Data

The dataset which has been taken for this research is from the github and its about the English-Hindi words pairs transliteration. However the data was still containing some non-English alphabets which could have degrade the learning the performance but this issue has been taken care of by removing the unwanted variables in python. The dataset consists of around total 14879 words pairs which were chosen randomly and converted into desired language. Out of these, 3719 words used as a test data and rest is used as a train set. Before that, while performing the experiment, the train data adjusted was about 8000 but due to less accuracy, the train data has been increased to the maximum.

4.3 Implementation of Decision Tree Model

Analysis, design, development and evaluation are the main components of the implementation. In the analysis process, after reading many research papers, the main goal was to find the best model about the transliteration for English-Hindi name pairs. English/Hindi word alignment, given a source language English and the phonetic equivalent in Hindi (Target Language), between them to find the most probable correspondence. Generally in English/Hindi alignment has some following properties like:

Null correspondence

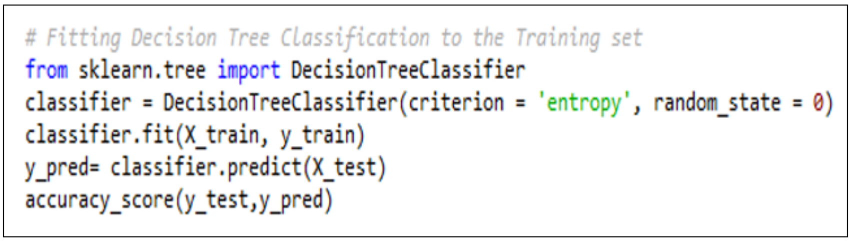
Many-to-many correspondence

No crossing dependency

In most of the cases, the lengths of English words and their Hindi transliteration are not same. Thus the mapping type should be many-to-many correspondences. Moreover it might be possible that it may have null correspondence. (Kang and Choi; 2000). The reason behind this is because of the silent English letters. Also when the correspondence are visualized as links between the English and Hindi words, the links do not cross each other. Thus there cannot be any effective alignment algorithm which is possible by drastically reducing the search space. In this project we have used decision trees model between the English and Hindi words so that the mapping can be happened easily. Moreover, in the source word side, the null pronunciation units makes the application difficult. Thus to sort out this problem, the alignment configuration needs to be constrained. Specifically, one-to-many correspondence will be allowed and in the source word side we will prohibit null pronunciation unit

The accuracy of transliteration and back transliteration is measured by the percentage of correctly transliterated words divided by the total number of words tested. Here, ID3 has also been used as it grows the trees until all the training examples are perfectly classified. It may lead to some difficulty if the training data contains some noise. This is because of decision trees tries to learn about the peculiar pattern including the target concept. So even though, on the training data if decision tree is performing well, it will perform poorly on an unseen data, thus leads to over fitting. Thus to avoid the over fitting, there are majorly two steps which are known as pre pruning and post pruning. To stop the tree before going to the perfect classification is known as pre pruning and deriving of the complete decision tree first and pruning it further is termed as post pruning.

As the dataset which have taken is a small dataset, but still in this research we have tried the best to get the alignment accuracy. However the accuracy could have been reached to the max if there should be a huge amount of data. Also in most cases , post pruning is known to be more effective than the pre pruning in case of avoiding the over fitting problem thus to get a persistent performance , a reduced error pruning method has been implemented so that we can find the best transliteration accuracy. As the dataset which we have is already aligned, thus for the decision trees induction, it is straight forward to generate the large training data. The below figure 5 shows about the implementation of decision trees in python:

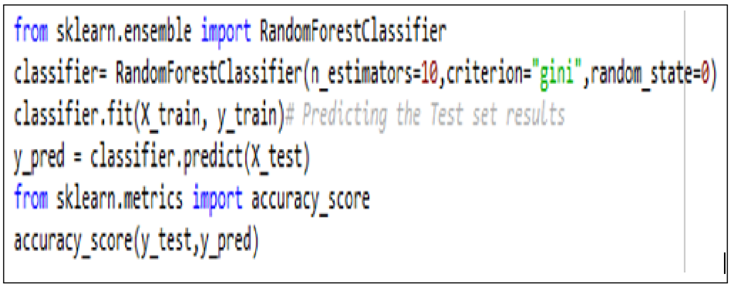


```
# Fitting Decision Tree Classification to the Training set
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)
y_pred= classifier.predict(X_test)
accuracy_score(y_test,y_pred)
```

Figure 5: Decision Tree

4.4 Implementation of Random Forest

Random forest is one of the best ensemble classification algorithm which is getting in trend for machine transliteration nowadays. In the ensemble classifier, instead of using the one classifier to predict the target variable, it uses the multiple classifier for the target prediction. In the random forest, these random classifiers are the randomly created decision trees. In general, to gain the higher accuracy, there should be higher number of trees in the forest. Some of the major advantages of the random forest includes the handling of the missing values and modelling of the random forest classifier for the categorical values (Mitchell et al.; 2011). In this project, random forest is used because we can create many trees in this data by which we can obtain the best and maximum accuracy for the transliteration of Hindi English word pairs. There has been very less researches which has been taken place regarding the Hindi to English transliteration, as per the previous work by the researchers which has happened in India for the Hindi transliteration, this model has been never used. The main motive of using the random forest here is because of the reduction in over fitting and chance of stumbling across the classifier which doesnt perform well because of the train and test data relationship can be reduced. The very first includes with training and testing of the data where the training data consist of the 14879 variables and test data consists of 3719 variables. The next step includes with the importing of the random forest classifier in python. The number of estimators which also means the number of trees in the forest was chosen default as 10 and the default string criterion was also chosen as default gini which is a function used to measure the quality of split. After that by putting the test data in the classifier and implementing it makes us to proceed to the next step which was importing of the accuracy score package. The below figure 6 is the implementation of Random forest in python:



```
from sklearn.ensemble import RandomForestClassifier
classifier= RandomForestClassifier(n_estimators=10,criterion="gini",random_state=0)
classifier.fit(X_train, y_train)# Predicting the Test set results
y_pred = classifier.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Figure 6: Random Forest

5 Evaluation and Results

5.1 Decision Tree

In past work on English/Hindi transliteration, for improving the mapping accuracy, pruning trees are turned out not to be very helpful. In this experiment, we have obtained the results that all the tree pruning methods failed to increase the transliteration accuracy, which means that ID3 is not over fitting the data. Also while performing this implementation, we have found that the tree size has been reduced after pre pruning and post pruning method but the accuracy remains the same. Until now the training data was consisting of about 10000 words, but here we have increased the size of the training data to get the better accuracy and avoid the over fitting of the data. The results which has been obtained was however not so good but at an average level. From the transliteration point of view, the accuracy which has been obtained was 39 percent is somewhat higher in terms of Hindi/English transliteration which has happened in India. There were very less researches which has been done on transliteration by using the Decision Tree algorithm. The main advantage which has been derived here by using Decision tree is that this algorithm requires comparatively less efforts among the other algorithm and also the nonlinear relationships between the variable does not affect the tree performance which means if there is any change in the attributes of data occurs then the model will not be effected with that. Also in future we there is a huge amount of data which can be used for the training and testing then to improve the performance of the model and the accuracy in the model, a hybrid model can be implemented.

5.2 Random Forest

While executing the random forest algorithm, we have observed that the results execution time which was taken by the random forest was very less as compared to the other two algorithm. Which derives us to one conclusion that the random forest doesnt not take too much time to train. Also the random forest model is fairly robust and requires very less need in terms of tuning of hyper-parameters. As we know that in case of transliteration of languages, the data which will be there consists of many variables as the words doesnt have a limit thus to derive the results with less time and efforts , random forest can be implemented. To make the Random Forest more interpretable, there is a very straight forward way which leads to a similar level of interpretability as linear models in a dynamic sense. As a sum of features contributions, every predictions can be trivially presented which shows that how the features leads to a particular predictions. The results which has been obtained here for transliteration of Hindi-English words pairs is 31 percent by using only 14879 words pairs. However, there is no such research have been made for the Transliteration of Indian languages using this method. The accuracy of the model can be increased to the maximum state if there are huge lists of words pairs available.

5.3 Comparision of Random Forest and Decision Tree

In this research, the model which has been applied for the transliteration are decision trees and random forest and each of them have performed really well in the experiment. If we talk about the random forest, it is like a black box and like a forest which one can build and control. The number of trees we want in the forest can be specified which is termed as n-estimators and also the number of features can be specified in each tree. But the randomness cant be controlled which means that we cant control which feature is a part of which tree . In our research in future if we get a big transliteration data consisting of many phonemes then it will not be easy for the random forest to predict that which data point is a part of which tree. But in case of using the decision tree model, the accuracy can be enhanced and increase if there are more and more number of splits. Also in case of large amount of data there can be a probability of over fitting but we can use the cross validation to adjust this property

6 Conclusion and Future Work

This research has represented a Machine transliteration for English-Hindi words pairs using two machine learning models. As English and Hindi are phonetically very rich languages thus phoneme is selected as a prime features. The models which has been applied here are Decision Tree and Random Forest. The method consists of a small dataset where the word alignment and predicting the accuracy was the major objective. Also we wanted to induce the transliteration rules for each Hindi and English words pairs. In the transliteration case, during the experiment it was found that the vowels of English has low letter accuracy and the reason is because the low accuracy of vowel is realized to many different phonemes. However the large labelled examples of the training set is what which was missing. Also there has been very less work which has been implemented on the English-Hindi words transliteration using decision trees and random forest and especially in India for Hindi language. In this research what we have found is how well a decision tree and random forest can work for transliteration if there is a large amount of data. In case of the large amount of data there will be more trees and thus more chances of accuracy can occur and also the prediction pace will be high which also includes the ease of coding as well.

As mentioned, if there is large amount data available for Hindi- English words pairs, then there is a huge chances of gaining the maximum the transliteration accuracy. Thus in future, we will be carrying forward this approach and parallely will try to apply the Xgboost technique which is an open source library which provides the gradient boosting. This approach can still be used for the typographical errors and spelling mistakes that occurs in the source language. Incorrect words are likely to be similar and can translate in the same way. Words to be considered for clustering with some method, a slight modification can be done and it would be possible to form clusters of words. Also for the best alignment of the words pairs, Hidden Markov Model will also be considered as hmm serves very well in terms for the words alignment.

7 Acknowledgement

I would like to thank my supervisor Dr Catherine Mulwa for her valuable advice, support, ideas, guidance and encouragement. Thank you for always having the time to help me and giving the direction to my work and for everything youve done for me over the course of Masters. I would like to thank my college, National College Of Ireland, especially NCI library for providing me different sources to gather information. Also special thanks to the github community for giving the suggestions towards the coding and technical part.

References

Antony, P. and Soman, K. (2011). Machine transliteration for indian languages: A literature survey, *International Journal of Scientific and Engineering Research* **2**(12).

Das, A., Ekbal, A., Mandal, T. and Bandyopadhyay, S. (2009). English to hindi machine transliteration system at news 2009, *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, Association for Computational Linguistics, pp. 80–83.

Emeneau, M. B. (1956). India as a lingustic area, *Language* **32**(1): 3–16.

Haizhou, L., Min, Z. and Jian, S. (2004). A joint source-channel model for machine transliteration, *Proceedings of the 42nd Annual Meeting on association for Computational Linguistics*, Association for Computational Linguistics, p. 159.

Kang, B.-J. and Choi, K.-S. (2000). Automatic transliteration and back-transliteration by decision tree learning., *LREC*, Citeseer.

Kang, I.-H. and Kim, G. (2000). English-to-korean transliteration using multiple unbounded overlapping phoneme chunks, *Proceedings of the 18th conference on Computational linguistics-Volume 1*, Association for Computational Linguistics, pp. 418–424.

Knight, K. and Graehl, J. (1998). Machine transliteration, *Computational linguistics* **24**(4): 599–612.

Mathur, S. and Saxena, V. P. (2014). Hybrid appraoch to english-hindi name entity transliteration, *Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students' Conference on*, IEEE, pp. 1–5.

Mitchell, L., Sloan, T. M., Mewissen, M., Ghazal, P., Forster, T., Piotrowski, M. and Trew, A. S. (2011). A parallel random forest classifier for r, *Proceedings of the second international workshop on Emerging computational methods for the life sciences*, ACM, pp. 1–6.

Oh, J., Choi, K. and Isahara, H. (2006a). A comparison of different machine transliteration models, *Journal of Artificial Intelligence Research* **27**: 119–151.

Oh, J., Choi, K. and Isahara, H. (2006b). A comparison of different machine transliteration models, *Journal of Artificial Intelligence Research* **27**: 119–151.

Oh, J., Choi, K. and Isahara, H. (2006c). A comparison of different machine transliteration models, *Journal of Artificial Intelligence Research* **27**: 119–151.

Pakray, P. and Bhaskar, P. (2013). Transliterated search system for indian languages, *Pre-proceedings of the 5th FIRE-2013 Workshop, Forum for Information Retrieval Evaluation (FIRE)*.

Rama, T., Singh, A. K. and Kolachina, S. (2009). Modeling letter-to-phoneme conversion as a phrase based statistical machine translation problem with minimum error rate training, *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, Association for Computational Linguistics, pp. 90–95.

Singh, A. K. (2006). A computational phonetic model for indian language scripts, *Constraints on Spelling Changes: Fifth International Workshop on Writing Systems*, Nijmegen, The Netherlands.

Sproat, R. (2003). A formal computational analysis of indic scripts, *International symposium on indic scripts: past and future, Tokyo.*

Stalls, B. G. and Knight, K. (1998). Translating names and technical terms in arabic text, *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, Association for Computational Linguistics, pp. 34–41.

Sunitha, C. and Jaya, A. (2015). A phoneme based model for english to malayalam transliteration, *Innovation Information in Computing Technologies (ICIICT), 2015 International Conference on*, IEEE, pp. 1–4.

Surana, H. and Singh, A. K. (2008). A more discerning and adaptable multilingual transliteration mechanism for indian languages, *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I.*

Teshome, M. G., Besacier, L., Taye, G. and Teferi, D. (2015). Phoneme-based english-amharic statistical machine translation, *AFRICON, 2015*, IEEE, pp. 1–5.

Wei, P. and Bo, X. (2008). Chinese-english transliteration using weighted finite-state transducers, *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, IEEE, pp. 1328–1333.