A Comparitive Analysis Multilingual Word Tagging in Transliterated Content

Yagnesh Jani[1] Sandip Modha[2]
[1]M.E Scholar [2]Assistant Professor
[1,2]Department of Computer Engineering
[1,2]KSV University, Gandhinagar, Gujarat, India

Abstract— In this report we consider the problem of labeling the languages of words in mixed-language documents and remove disambiguity. For many languages that use non-Roman based indigenousscripts (e.g., Arabic, Greek and Indic languages) one can often find a large amount of user generated transliterated content on the Web in the Roman script. This content createsa monolingual or multi-lingual space with much script which we refer to as the Transliterated-Script space. This report describes a method to tag the word in multilingual transliterated content and remove disambiguity. The technique is based on the heuristic approach and comparing to different methods results. Therefore, it is a non-trivial task to tag the word and find disambiguity.

***Key words*:** transliteration, devnagari script, roman script

I. INTRODUCTION

Many languages, including Arabic, French, and most of the South andSouth East Asian languages, are written using roman scripts. More often the web content and the user generated search (such as tweets and blogs and queries) in these languagesare written using Roman script because of various socio-cultural and tools reasons. This process of phonetically representing the words of a language in a nonnative script is called transliteration [7]. A lack of standard keyboards, a large numberof scripts, as well as popularity with English and U.S. keyboards has given riseto a number of transliteration schemes that are used for generating Indian language text in Roman transliteration. Transliteration, mainly into Roman script, is used more often on the Web contents, user queries that intend to search for related transliterate documents. Songs, etc. search engine face challenges while processing transliterated queries anddocuments is that of large spelling variation. For instance, the word Pahala ("First" in Hindi and many other Indianlanguages) can be written in Roman script as pahalaa, pehla, pahila, pehlaa, pehala,pehalaa, pahela, pahlaa, pahelo etc. So identifying this type of words in such transliterated documents is real time task. Another problem is there are ambiguous word in Hindi and English langue. For instance, the word MAN ("MIND" in Hindi langue and "PERSON" in English). So find these ambiguous words in document and queries and derive whether it is belongs to Hindi or English langue is another problem. This problem has mainly two task.

- Query word labeling
- Try to remove disambiguity

This is the problem of matching of non-trivial term for search engine to match transliterated query with web content with spelling variation. And we can find ambiguous word in that query. So it has mainly two task

- Handle spelling variation in same script
- Remove disambiguity

II. TASK DEFINITION

A query q: <w1w2w3w4…. wn> or twits can be written in roman script. The words of query could be in English language or in transliterated from Devnagari languages ex. Hindi. The object of the task is to identify the word as it come from English language or it come from transliterated Devnagari Hindi language. We can find one challenge in this task that is disambiguity. Some time we can find word which can be in both English and transliterated language ex. (MAN can be meaning as person in English and mind in transliterated Hindi language). So system has to 1st identify that ambiguous words and after that it can perform tagging by concluding that meaning.

III. DATASETS

This section describe the dataset which is used in this task. The training and test data is given by FIRE and in this dataset some random chat of different user are given. Even this data sets some English word frequency list and Hindi word frequency list is given. Moreover our guide gives some real time twits data which is in transliterated form more likely 300 twits which we can use in our future work. In different user chat there are Hindi word transliterated pair which we can use as data set dictionary for our system.

IV. SYSTEM DESCRIPTION

We divide the overall task in mainly two problems, (a) word level tagine (b) disambiguity remove

A. Query Word Labeling:

From few studies we can successfully deduced that n-gram character model with statistic approach obtained reasonable results. There for we used unigram up to trigram model. Ere we use word gram model instead of character gram model. Our approach is simple and heuristic. In our system we 1st take query as one sentence after that we do cleaning on that input query. Cleaning process in like remove symbols ex. #, @, or any numbers. After that we can generate tokens of single word foe unigram or 2 words pair for bigram model and store that words in one array.

Now we have array so we can check that particular word in English dictionary statically which is available on internet or we can use some google dictionary api to check whether this word is available in English dictionary or not. On this stage we bifurcate words which are available in English dictionary.

- Se <W_1 W_4 W_5…. W_n> Set of words which are found in English dictionary.
- Sh <W_2 W_3 $W_{6\ldots}$ Wn >. Remaining words which were tags as H (Hindi word).

Now we use Se set and again all words of this sets can be compare with transliterated data sets which is given

by FIRE. This data set contain transliterated word pairs of Hindi-English words pairs. If we can found any word in this dataset than we remove that word from Se set and sore in new array which is ambiguous words array. So by this step we get ambiguous words. Other words remaining in set Se are tagged as E (English words). By this approach we get particular Hindi words and English words without ambiguous words. Steps are as follows.

1) Step 1: Cleaning given input query.
2) Step 2: Generate token for input.
3) Step 3: Match the word in English dictionary.
4) Step 4: Separate the matched English word in different array and other words in different array
5) Step 5: Now match that English word in transliterated data set given by FIRE.
6) Step 6: If any word found it is ambiguous word.
7) Step 7: separate that word from English word array and tag it as ambiguous word and

Other remaining words are tagged as E if in English array or as H if in Hindi array list.

B. Disambiguity Removel:

In this task we use set of ambiguous words which we can found in previous task. For this we use probabilistic method of n-gram model.

- P ($W_{1)}$ = P (W_1 / W_2) for unigram
- P ($W_{2)}$ = P (W_2 / W_1 W_3) for bigram
- P (W_3) = P (W_2 / W_1 W_3 W_4) for trigram

We use these equations for different model. Foe probabilistic approach we use neighbor of ambiguous word. If more Hindi words are available in neighbor then we conclude that ambiguous word as Hindi word and if more English words are available than we conclude that ambiguous word as English word.

Here after some runs we can see that as we goes from unigram to trigram we obtain more accuracy in results. Here we just goes for trigram model only. We also compare our results of 1st task with result of hmm model. So with this approach we can try to remove disambiguity in queries or documents.

V. Results and Future Work

- Precision is the number of correct results divided by the number of all returned results.
- Recall is the number of correct results divided by the number of results that should have been returned
- F-measure

$$F = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

A. Subtask 1:

1) Our Approach:

language	Precision	Recall	F-measure
English	0.733	0.826	0.776
Hindi	0.764	0.793	0.778

Table 1:

2) HMM Model:

language	Precision	Recall	F-measure
English	0.920	0.843	0.880
Hindi	0.922	0.843	0.881

Table 2: [5]

B. Subtask 2:

1) For Disambiguity:

Language	Precision	Recall	F-measure
English	0.792	0.723	0.755
Hindi	0.682	0.652	0.666

Table 3:

VI. Conclusion Future Work

From our task and implementation we can conclude that as if we goes for 4 gram to more n-gran model we can enhance our system result and from result we can see that hmm model gives us much batter result for this task.

We can extend our work for real time application like twitter. We will try to implement this work for enhance search engine.

Reference

[1] DCU@FIRE-2014: Fuzzy Queries with Rule-based Normalization for Mixed Script Information Retrieval by Debasis Ganguly, Gareth J. F. Jones in fire conference 2015 Bangalore.
[2] LIGA and Syllabification Approach for Language Identification and Back Transliteration: Shared Task Report by DA-IICT by Shraddha Patel, Vaibhavi Desai in fire conference 2015 Bangalore.
[3] Query Word Labelling and Transliteration for Indian Languages: IITP TS Shared Task system description by Shubham Kumar, Deepak Kumar Gupta, Dr. Asif Ekbal in fire conference 2015 Bangalore.
[4] A Hybrid Approach for Transliterated Word-Level Language Identification: CRF with Post Processing Heuristics by Somnath Banerjee+, Aniruddha Roy+, Alapan Kuila+, Sudip Kumar Naskar+, Sivaji Bandyopadhyay+, and Paolo Rosso# in fire conference 2015 Bangalore.
[5] Machine learning approach for language identification and transliteration: shared task reort of IITP-TS by Deepak kumar guota, Subham kumar, Asif Ekbal in fire conference 2015 Bangalore.

