ISM@FIRE-2013 shared task on Transliterated Search

Dinesh Kumar Prabhakar, Sukomal Pal

Indian School of Mines, Dhanbad,

Jharkhand, India

{dinesh.nitr, sukomalpal}@gmail.com

Abstract. This paper describes the work that we did for FIRE 2013 shared task on Transliterated Search at ISM Dhanbad. The technique solves the problem of labeling the languages, English or Hindi (E/H) of query words in mixed-language sentences using spelling based techniques. We are participating only in Subtask 1: Query Word Labeling, where we have worked based on English-Hindi (E-H) transliteration pairs. From the training data we made 26 different files for English words and one single file for Hindi words. For the given test data, we tried to find a match first in English files and then in Hindi file for each token of the input line. This simple algorithm performs quite well for exact query match fraction, Transliteration-precision and Transliteration-recall but does not do up to the mark for new words, not present in training data.

Keywords: Transliteration, Language labeling

1 Introduction

A large number of languages are written using indigenous scripts. Several language comes under this scripts including South and South Asian language. But people write the content languages in Roman script due to various socio-cultural and technological reasons. This process of phonetically representing the words of a language in a non-native script is called *transliteration (forward)*. Transliteration is used frequently not only on web for documents, but also for user queries that intend to search for these documents.

1.1 Task Description

This year, there were two subtasks hosted in shared task on Transliterated Search: first a query labeling task, and the second is ad hoc retrieval task for Hindi film lyrics. The description of the subtask-1 is given below for which we have submitted our run.

Subtask 1: Query Word Labeling. Assume that *q*: w_1, w_2, w_3 … w_n is a query is written in Roman script. The words,w_1,w_2 etc., could be standard English words or transliterated from another language **L,** in our case it is **H**. One more thing is that if the word is name of the person and places in **L**, can be skipped during labeling.

Next we have discussed labeling related work in Section 2. Section 3 discusses our methodology, Section 4 analysis of our work, and finally we conclude in Section 5.

2 Related Work

There are some sequence tagging tools available like MALLET(A Machine Learning for Language Toolkit) Sequence Tagger. This tagger can be used for classification of text. Ben King and Steven Abney have only worked in labeling the languages of words in mixed-language documents[1]. They have approached this problem in a weakly supervised fashion, as a sequence labeling problem with monolingual text samples for training data. Among the approaches evaluated,they found a conditional random field(CRF) model trained with generalized expectation criteria was the most accurate and performed consistently since the amount of training data was varied.

3 Our Approach

In our approach we have designed our system purely on lookup based. We have used the data-set of 30824 English-Hindi pairs words[2], 207856 English word with frequency[3].

Based on first character, these english words are splitted and stored into 26 different files. Initially system reads a term (*word*) from text file and compares first character of *word* with files name, and then search the *word* in matched file (e.g. suppose *Apple* is input word and first character is *A*, so this term will be searched in file *a.txt*).

If word found in respective english file we temporarily assume it is english term , but there is possibility that it may be a Hindi term. For this reason without taking final decision we search term in E-H pair file. If the term is found here also we assume it may be hindi term. Hence we assign H to word as label.

There are some out-of-dictionary terms are typically names, such as companies, people, places, and products [5] for these terms transliteration in native script need not require in our language labeling task. This is purely lookup based system which searches the input related query if found give the the related result based on search.

We have design an algorithm based on which our system works is given below.

Algorithm

```
1. Input term from Test Document
2. Check first letter of  word {A-Z,a-z}
3. Search  word in corresponding Document
4. if  match found
4.1.    { search  word in E-H pair Document
4.2.      if found
4.2.1.        {print word,\H, word's native script from E-H pair}
4.3.              else
4.3.1                 {print word,\E}}
5.  else
5.1.    {Search in E-H pair Document
5.2.      if found
5.2.1.    {Print word ,\H,=, native script from E-H pair}
5.3.      else
5.3.1.       {Print  word, \H}}
6. end
```

4 Analysis

Relative scores of various metrics for our runs are mentioned in Table 1 along with Maximum and Median scores.

The FIRE forum have used the following metrics for evaluating Subtask 1: Exact query match fraction, Exact transliteration pair match, Transliteration-precision, Transliteration-recall, Transliteration-F-score, Labeling accuracy, Eng-precision and Eng-recall and similarly L-precision, L-recall and L-F-score, where L is H (Hindi) in our case.

As we can observe in the Table 1, for some of the metrics namely Exact query match fraction, Transliteration-precision, Transliteration-recall and Transliteration-F-score we perform better than Median Scores. For the other metrics, our scores are, however lower but near to Median Scores.

Table 1. Relative scores of various matrics

Language Stats	Metrics	ISMDhanbad Score	Maximum Score	Median Score
Hindi	Exact query match fraction	0.0860	0.1980	0.0290
10 runs	Exact transliteration pairs match	1584/2117	N. A.	N. A.
5 teams	Transliteration -precision	0.7253	0.8135	0.4486
#(True \H) = 2444	Transliteration -recall	0.6484	0.8125	0.4300
#(True \E) = 777	Transliteration -Fscore	0.6847	0.8130	0.4260
#(\N)= 232	Labelling accuracy	0.8780	0.9848	0.9540
N = Names	Eng-precision	0.6853	0.9667	0.9302
ambiguities	Eng-recall	0.9138	0.9755	0.9640
excluded from	Eng-Fscore	0.7832	0.9685	0.9019
analysis	H-precision	0.9693	0.9906	0.9883
	H-recall	0.8666	0.9894	0.9791
	H-Fscore	0.9151	0.9900	0.9700

Since our technique is purely look-up based, we can not properly handle the new words not present in the training data. We have also failed in correctly identifying named entities (NE). Possibly use of some NER tool can solve this problem. On a different note, there were some errors in the transliteration-pairs of training data. Since our system is heavily biased by the training data, we got inexact transliteration for some terms.

5 Conclusion

Our work is based on spelling based backward transliteration technique. As we have used English-Hindi transliteration pairs for transliterated search and English word list for classifying the English and Hindi terms. Our system has performed better for some of metric considered to evaluate the system output. However there are some limitations of this system. System may not give appropriate labeling we haven't used any NER technique to identify the named entity. Further we are planning to improve result of our system.

References

1. King, B., Abney, S.: Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods In Proceedings of NAACL-HLT-2013, Atlanta, Georgia (2013) 1110-1119

2. Gupta, K., Choudhury, M., and Bali, K.: Mining Hindi-English Transliteration Pairs from Online Hindi Lyrics, In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC '12), Istanbul, Turkey (2012) 2459-2465

3. Sowmya, V.B., Choudhury, M., Bali K., Dasgupta, T. and Basu, A.: Resource Creation for Training and Testing of Transliteration Systems for Indian Languages, LREC (2010)

4. Karimi, S., Scholer F., and Turpin, A.: Machine Transliteration Survey. In ACM Computing Surveys (CSUR), Volume 43 Issue 3, New York, USA (2011) 17:1-46

5. DALE, R.: Language Technology. Slides of HCSNet Summer School Course. Sydney (2007)