COMP90049 Project 1 Report

Alisha Aneja

1. Introduction

This report aims at analysing and comparing the efficiency of various approximate string matching methods, namely Levenshtein Distance, Jaccard index and a combination of BLOSUM matrix and Soundex algorithm (as an extension) for the purpose of back-transliteration of names in Persian script to the original Latin word being represented. We are provided with two files, names.txt, which has a list of about 26K lowercased names in Latin script and train.txt, which has about 13K names in the Persian script with their lowercases equivalent in the Latin script.

2. Approximate Matching

The following methods are used in the project to find the 'best match' for a 'misspelled' Persian name from the dictionary of Latin names.

2.1. Levenshtein Distance

Levenshtein Distance measures the distance between a Persian name and the Latin names in the dictionary and consequently selects the Latin names with the minimum score. This minimum score represents the minimum number of single charater edits (insertions, deletions and substitutions). The [m,i,d,r] parameter here is [0, -1,-1,-1].

2.1.1. Evaluation Metrics

Recall ~ 45%

Precision ~ 4.6%

2.1.2. Analysis

Recall value of 45% tells that this method is able to predict correctly 45% of the total values, i.e. it predicted the correct Latin name for approximately 7K Persian words. But a precision value of 4.6% tells that for many Persian names, the method predicted a lot of Latin names., i.e., it isn't specific enough.

For illustration,

Persian Word → MDBA

Latin Word Predictions by the method → "aba","ada","alba","ba","baba","cuba","dea","dua","edda","edna","elba","gaba","haba","ida","madaba","maha","mama","mana","mara","mata","maya","mb","md","mds","medea","media","melba","mesa","mia","mika","mina","mira","mona","mura","musa","myra","raba","reba" .

38 Possible Latin name matches!

There will definitely be a requirement of application of more knowledge technologies because there are a lot of distinct paths that wind up at the same state.

Moreover, it doesn't take into account the distance of one letter from another letter, i.e., misspellings of a word are more likely to occur at the endings of the word or more so a letter can be mistakenly typed with a letter which is just besides it on the keyboard.

Also, it doesn't considers the phonetics of the Persian word or phonological properties of the source language, i.e., Persian.

For illustration,

' (single apostrophe) in Persian language cannot be translated back to Latin language and neither this method takes it into account.

Persian Word → JSYKA sounds very similar to Latin Word → Jessica

But the method is not able to determine it as it's best match because phonetics sure is not it's forte!

The cases where it is able to predict the correct Latin name are mostly where the orthographies of both the Latin and Persian word are same.

For illustration,

Persian Word → ABA

Latin Word → aba

Also, where due to translation between two languages, the vowels lose their meaning, this method passes the bar!

For illustration,

Persian name → QDAMH

Latin name → qudamah

Persian name → TNTS

Latin name → tants

2.2. Jaccord Index

Jaccord's coefficient calculates the similarity between the Persian and Latin name by calculating the fraction of similar letters in both the words.

2.2.1. Evaluation Metrics

Recall ~ 15.5%

Precision ~ 4.8%

2.1.2. Analysis

Recall value of 15.5% tells that this method is able to predict correctly 15.5% of the total values, i.e. it predicted the correct Latin name for approximately 2K Persian words, which is quite less. A precision value of 4.8% tells that for many Persian names, the method predicted a lot of Latin names., i.e., it isn't specific enough.

Also, it's time complexity is great and hence, performance is weak. But, it is a 'true' distance as opposed to Levenshtein distance and takes into account the relative ordering of the letters in the word; because 'matches' can be anywhere in the word as phonetics of the source language, i.e., Persian is different than the Latin language. It considers set of all rotations.

For illustration,

Persian name → RVSNFLD

Latin name → rosenfeld

Here, n,v,f,l,d are not at the same places in both the words. So, comparing the overall similarity of both the words will give better results, hence Jaccord coefficient proves useful here.

3. Extension of project – Combining the concept of Soundex Method and BLOSUM Matrix and Modification of [m,i,d,r]

Soundex method assignes the same scores to the letters with similar phonetics. Using this concept, the BLOSUM matrix in this project is derived. On replacement, all letters sounding similar are assigned a score of 1, else they are assigned a random score between 2 and 9. On insertion and deletion, a score of -1 is assigned. This way, it makes sure, phonetics take a lead over orthography, while matching Latin word name with Persian name.

```
vowels = ['a','e','h','i','o','u','w','y']
labials = ['b','p','f','v']
misc = ['c','g','j','k','q','s','x','z']
dentals = ['d','t']
lateral = ['l']
nasal = ['m','n']
rhotic = ['r']
```

3.1. Evaluation Metrics

Recall ~ 28%

Precision ~ 7%

3.2. Comparison with Levenshtein Distance and Jaccord Index

Recall value of 28% tells that this method is able to predict correctly 28% of the total Persian names, i.e., approximately 4K, which is better than Jaccord Index but worse than Levenshtein Distance and a precision value of 7% is better than both of the other methods.

	Levenshtein Distance	Jaccord Index	Soundex+BLOSUM
Recall	45%	15.5%	28%
Precision	4.6%	4.8%	7%

Table 3.2.1 Comparison of the methods using evaluation metrics

Sibilants often have overlap within the word. While translation from Persian to Latin, there is no clear orthographic distinction, that can be due to different mappings like vowel length. They are likely to be phonologically connected.

For illustration,

Persian name → ASTVNSN

Latin name → stevenson

This is correctly predicted by this method,whereas the other two methods were not able to predict it. It is clearly visible that the phonetics of both the words is similar.

For illustration,

Persian name → RVSNFLD

Latin name → rosenfeld

Persian name → TKYAH

Latin name → takiyah

All the three methods correctly predicted the Latin name but this method just predicted these single 'best match' whereas other two methods gave more than four 'best matches'.

But, it cannot work in cases where the phonology and orthography is entirely different.

For illustration,

Persian word → ASTVN

Predicted Latin word → aston

Correct Latin word → upston

3. Conclusions

According to the analysis, Levenshtein Distance gave the best recall value whereas Soundex plus BLOSUM method gave the best precision value. But, back-transliteration, is a challenging problem to do as a simple spell checker. The approximate string search methods cannot accurately perform search for acronyms. Code-mixing, i.e., interpersing of one language words with other language words is another challenge. Misspellings like typing errors are very common, because human do make mistakes! The main challenge is of phonological variations because phonology and orthography are at the same time independent of each other as well as inter-related. But, to really address these issues, system with special mechanisms should be designed using human 'knowledge'!

References

Sarvnaz Karimi, Andrew Turpin, and Falk Scholer (2006) English to Persian Transliteration. In Proceedings of the 13th Symposium on String Processing and Information Retrieval (SPIRE'06), Glasgow, UK, pp. 255–266.

Sarvnaz Karimi, Andrew Turpin, and Falk Scholer (2007) Corpus Effects on the Evaluation of Automated Transliteration Systems. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07), Prague, Czech Republic, pp. 640–647.

Umair Z Ahmed, Kalika Bali, Monojit Choudhury, Sowmya VB, Microsoft Research Labs India,Bangalore(2011),Challenges in Designing Input Method Editors for Indian Languages: The Role of Word-Origin and Context.

Justin Zobel, Philip Dart, Phonetic String Matching: Lessons from Information Retrieval