Knowledge Technology Assignment1 Semester1

Xiaodong Han 822581

1 Introduction

2 Data-sets

All resource data-sets (names.txt, train.txt) used in this report are given by Karimi et al. (2006). File names.txt contains a bunch of Latin names, which will probably be the dictionary; file train.txt contains a bunch of Persian names with their (correct) Latin equivalent.

3 Global Edit Distance

In this section, calculating the different GED between each (uppercase) Persian name in train.txt and all (lowercase) Latin names in names.txt. The best-scoring Latin name(s) from names.txt is/are the prediction(s), which can be compared with the true (lowercase) Latin name in train.txt to verify whether it is correct. In particular, using different parameter will return different result.

3.1 Using Basic Parameter

Using parameter [m, i, d, r] = [+1, -1, -1, -1], which is a basic parameter.The performance is as follow: (see Table 1).

Metric	Proportion
Precision	6207/43056=14.416%
Recall	6207/13437=46.193%

Table 1: Performance of Global Edit Distance

Table1 shows the performance of using basic parameter, but the precision in this table is not ideal. This is because basic parameter only considers the numbers of edit operations without taking into account the fact that the same operation for different character pairs has different weights and even different operations for same character pair have different weight. For example, for same character pair K and c, replacement may have higher weight than delectation, since K and c have similar pronunciation. Thus, it is not accurate to give all characters a same operation weight.

3.2 Modifying 'r' Parameter

Based on the observation of train.txt, I find that, in the process of transliteration between Persian name and Latin name, there are a lot of replacements between different characters which have the same pronunciation in English. I assume that the pronunciation of Persian name and Latin name follows English pronunciation rules, so that I can use Soundex code rule to divide all characters into six groups and ignore character ' which appears very few times. Thus, when calculating the distance, if two characters are involved in the same group, then the replacement counts 0, whereas replacement counts -1. Based on this rule, I modify a new parameter, which is [m, i, d, r] = [+1, -1, -1, (-1, 0)]. The performance of using new parameter is as follow: (see Table 2).

Metric	Proportion
Precision	7342/44531=16.487%
Recall	7342/13437=54.640%

Table 2: Performance of Global Edit Distance

This performance shows that both precision and recall are increased, which means the number of incorrect names returned is decreased and the number of correct names is increased. There are two main discoveries based on this improvement. First of all, in the process of transliteration of Persian name and Latin name, characters having the similar pronunciation have the large proportion of being replaced. Secondly, Persian and Latin may have the same pronunciation with English. Taking Persian name ABDVN as an example, the exact Latin name is ebdon. Using GED with basic parameter, the best-scored match is labdon, in this case, the accuracy of matching this word is 0%. While us-

ing GED with new parameter, the best-scored matches are ebdon and labdon, and the accuracy is 50%.

In order to further improve efficiency, I calculate the frequency of characters appearing in Latin name, see Figure 1.

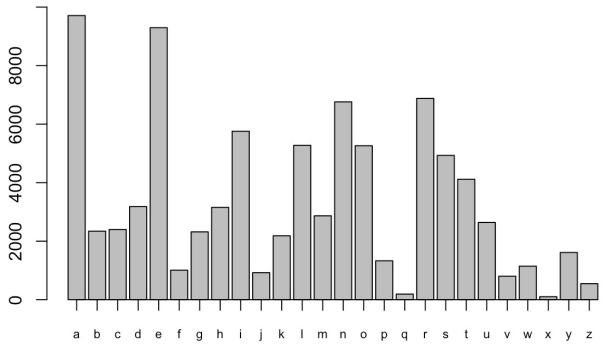

Figure 1: Frequency of characters appearing in Latin names

As it shows in the 1, based on the Soundex code rules, the average frequency of characters in group (a, e, h, i, o, u, w, y) appearing in Latin name is larger than other groups. That means characters in group (a, e, h, i, o, u, w, y) have higher possibility appearing in Latin name. Thus, when calculating the distance, if two characters are involved in group (a, e, h, i, o, u, w, y), then the replacement counts 1, while if two characters are involved in the same group but not group (a, e, h, i, o, u, w, y), then the replacement counts 0, and in other cases, the replacement counts -1. The new parameter is [m, i, d, r] = [+2, -1, -1, (-1, 0, 1)], the performance of this parameter is as follow: (see Table 3).

Metric	Proportion
Precision	7439/31044=23.963%
Recall	7439/13437=55.362%

Table 3: Performance of Global Edit Distance

Different with second modified parameter, this one decreases the number of incorrect names, which has increased the precision of this system. For example, the exact Latin name which matching the Persian name "AGLSTVN" is "eaglestone"; for GED with basic parameter, it only care about operation times; for GED with second modified parameter, it does not distinguish the different weightes towards different Soundex-code-group; so that only GED with third modified parameter can predict this kind of words.

4 N-Gram

Using N-Gram, the performance of using different n parameter is as follow: (see Table 5).

Metric	Precision & Recall
N = 2	3513/42970=8.175 & 3513/13437=26.144
N = 3	2674/34959=7.649 & 2674/13437=19.900
N = 4	2264/35091=6.452 & 2264/13437=16.849

Table 4: Performance of N-Gram

The precision gets lower with increasing 'n', which shows that N-Gram does not work well in approaching this problem. Based on the observation of train.txt, it is easy to say that there are a lot of differences between Persian name and Latin name, since they belong to different language script. Thus, Persian name and Latin name have less same substring, so that the number of correct match will decrease with the increase of n.

5 Local Edit Distance

Using Local Edit Distance, the performance is as follow: (see Table 5).

Metric	Proportion
Precision	5626/983322=0.572 %
Recall	5626/13437=41.869%

Table 5: Performance of Local Edit Distance

Comparing with the performance of GED, it is clearly shown in 5 that using LED get quite lower precision, which shows that LED has returned too many incorrect Latin names. The main reason is that LED is used to calculate distance to verify which two strings have the most similar substring, which means a high score of two words shows that they have the most similar substring instead of they are most similar. For example, there are more than 15 Latin names in name.txt, like canada, grenada, quesada etc, have matched the Persian name ABADA, since every matched name has a substring aba. In this case, every Latin name which has the most

similar substring with Persian name will be returned to users, thus this method is not applied to this problem.

6 Conclusions

In conclusion, LED and N-Gram algorithm is not useful for this approximate match problem. As for other methods, all of them have higher recall than their own precision, in this project, that means many irrelevant names have been predicted. Thus, the problem of using approximate match method to perform backtransliteration is to find an appropriate method or improve existing methods to improve the precision without lowering recall. In particular, modifing a special r parameter when using GED.

References

Sarvnaz Karimi, Andrew Turpin, and Falk Scholer. 2006. English to persian transliteration. *Proceedings of the 13th Symposium on String Processing and Information Retrieval (SPIRE06)*, 4209:255–266.