COMP90049 Project 1 Report: Whats in a (Persian) name?

1 Introduction

The aim of this report is implement a Global Edit Distance(herein GED) based prediction system in order to find correct "back-transliterations", i.e. to find the original word before another system transliterated it. The initial system will rely on a basic GED implementation, and, through knowledge gained from analysis of this implementation, a tuned system will replace it.

2 Dataset

The data involves foreign words that have been transliterated into the Persian script by Karimi et al. (2006) and Karimi et al. (2007). They have then been transcoded into the Latinate equivalent using the DIN 31635 standard[1]. Characters that have multiple representations have been transcoded with the same Latin character each time. The dataset comprises of 11967 Persian transliterations with the original foreign name given alongside in tab-delimited format. Another dataset of 25909 Latin names was also supplied, from which we are to source our predictions when creating and testing our model.

3 Evaluation Metrics

Throughout this paper, the following terms will be used to evaluate each system:

- Accuracy: For systems that predict only one name, which proportion are correct, or, for systems that predict more than one name, which proportion of Persian names have at least one correct answer
- Precision: For systems that predict more than one name, the average proportion that are correct

The runtime is also shown, as different situations may call for quicker results at the cost of accuracy. The evaluating metric, however, does not take this into account, and it is shown merely for informed comparison in case of real-world deployment. Finally, in the case of multi-prediction systems, the average number of predictions returned, and the maximum number of predictions is also given.

Unless specified otherwise, all testing was done on the same subset of 500 Persian names in the interest of speed and to avoid over-fitting the system.

[1]http://transliteration.eki.ee/pdf/Arabic_2.2.pdf

4 Original Global Edit Distance

4.1 Basic

The initial GED used the Levenshtein parameters of $(m, i, d, r) = (0, 1, 1, 1)$, and the editdistance library from Hiroyuki Tanaka (2013), which is written in C++ with a python wrapper for speed. The results are summarised in Table 1 :

Accuracy	0.454
Precision	0.300
Avg. Predictions	7.94
Max. Predictions	156
Time/s	23.1

Table 1: Results of basic GED for 500 names

4.2 Reduced Predictions

As can be seen above, the system is relatively accurate at returning a correct original name, but by averaging almost 8 predictions for each Persian name, this is not a very precise system. In order to only return one prediction, the candidates were then ranked using the n-gram distance (using Graham Poulter (2012)), and the best result was returned. For all other multi-prediction systems, the same method will be used, unless specified otherwise. The results of this modification are shown in Table 2:

Accuracy	0.286
Time/s	23.2

Table 2: Results of reduced GED

4.3 Analysis

The Levenshtein distance is effective in situations where transliteration then transcoding has not substituted the original letters, only removed some. A few examples where this is evident in the correct predictions in Table 3. The system was less successful however, when the processed changed the original letters

Persian	**Prediction**	**Correct**
RNZLND	renzland	✓
GRNANDR	grenander	✓
BSAK	bosak	✓
NYYVKAMB	kamb	newcomb
VANDNBRQ	vandry	vandenburgh
KAVDLYNG	katelyn	quodling
GRASR	graser	grosser
BYSMARK	bismark	bismarck
BRNDYS	brandys	brandeis
DYLYNGR	dylan	dellinger

Table 3: Examples of reduced GED

5 Modifications to Global Edit Distance

5.1 Insights

We investigated the following categories and made modifications based on them.

5.1.1 Transliteration Mappings

Examining the training dataset, and looking at the changes that transliteration and transcoding makes, there are a few obvious modifications. The character *x* is almost exclusively represented as *ks*, similarly *q*, if not represented as *q*, is instead *k*, and *'* maps almost exclusively to *a*. The first modification is to change how the "match" of the GED algorithm is computed. This is done by saying a *k* in the Persian name matches both *k* and *q* in the English candidate and *'* similarly matches *a*. It is more difficult to deal with *ks*, as it is two characters. Our solution is to calculate the GED for both the original string and the string with *x* substituted in for *ks*, and to take the best result.

Transliteration often results in a shorter word. The average length of the Persian words is 5.44 compared to the English 6.45. There are two categories of characters that are removed, repeated characters, and vowels, in particular *'e*. By comparing the Persian and English with the python snippet:

```
for tk, en in train_all.items():
  m = re.match(r".*(.)\1.*", tk)
  if m is not None:
    c = m.group(0)[0]
    m_str = r".*{}{}.*".format(c,c)
    if re.match(m_str, en) is not None:
      print(tk, en)
```

We observed that the only transliterations to preserve the correct double letters are *aamina*→*aamyna* and *aaron*→*aarvn*. Therefore, before the edit distance is calculated, all English repeated characters are reduced to one character. The loss of vowels is accounted for by changing the weighting of the *i* parameter to 0 for *e* and 0.5 for other vowels.

5.1.2 Transcoding Mappings

The transcoding with the DIN 3165 standard presents another area where examination provides benefits. The transcoding uses the same mapping for each Persian character, but two characters in particular, *vāv* and *ye*, map to multiple English characters. To account for this, *v*, *o*, *u*, and *w* are all said to match each other, as are *y* and *i*.

5.1.3 Pronunciation

A few minor common similarities in pronunciation are also accounted for in a similar way as in section 5.1.2. These are: *c* and *k*, and *z* and *s*.

5.1.4 GED Parameter Variation

Finally, the actual (m, i, d, r) parameters of the GED were varied. The usual context of edit distance for NLP is spelling correction. This is quite different to back-transliteration, and therefore the parameters are adjusted accordingly. Matching is far more important, as transliterations shouldn't changed that beyond what is outlined above. Deleting is also far more serious, as there should very rarely be the case where deleting is required. After evaluating the system with multiple parameters, we went with (-5, 1, 8, 1).

5.2 Implementation

The code uses a modified version of the damerau_levenshtein function from Daniel Lindsley (2014), then optimised in cython. This is orders of magnitude slower, but the modularity desired requires a pure python algorithm. The n-gram calculation is similarly modified for the reduced version to account for the insights above.

5.3 Results

The results for both the modified GED and the reduced version are shown below:

Accuracy	0.732
Precision	0.600
Avg. Predictions	1.78
Max. Predictions	16
Time/s	584

Table 4: Results of modified GED for 500 names

Accuracy	0.598
Time/s	585

Table 5: Results of reduced modified GED

5.4 Analysis

The modification is markedly more successful than the original. It is able to correctly predict most names where the basic GED failed. Where it still falls short, however, surnames with spelling variations. This is particularly evident in some of the incorrect predictions in Table 6, where the prediction is another variant of the last name. It is, however, much slower than the basic GED.

Persian	Prediction	Correct
RNZLND	renzland	✓
GRNANDR	grenander	✓
BSAK	bosak	✓
NYYVKAMB	newcomb	✓
VANDNBRQ	vandenburgh	✓
KAVDLYNG	quodling	✓
GRASR	graser	grosser
BYSMARK	bismark	bismarck
BRNDYS	brandys	brandeis
DYLYNGR	dillinger	dellinger

Table 6: Examples of reduced modified GED

6 Phonetic Algorithm

Considering the problem of transliteration, the issue is less one of spelling similarity, and more one of phonetic similarity. Therefore, we have

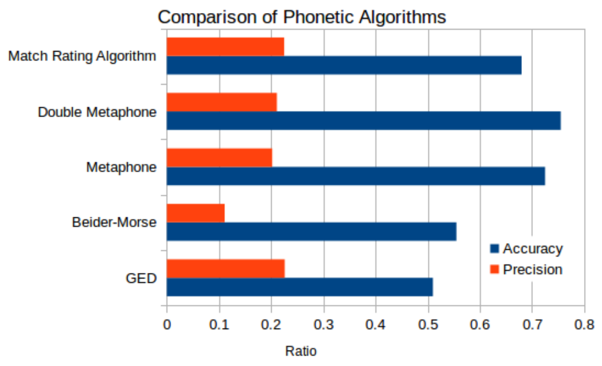

Figure 1: Phonetic Algorithm Comparison

also investigated a number of phonetic algorithms from Chris Little (2014 2015), and their relative performance. The aim was to use a phonetic algorithm to reduce the prediction candidates, then run the modified GED on the result in order to gain the best prediction.

6.1 Methodology

The testing was done on the same subset of data. Accuracy was the most important metric, as further filtering would be done by the GED/n-gram model. Details to note about the implementation are:

- Both the Persian and the English are run through the phonetic algorithm and the output is compared by the base Levenshtein. Levenshtein is used instead of the modified GED as it is more suited to the phonetic algorithms' output.
- The Beider-Morse algorithm was run with the English words described as English and the Persian as Arabic.
- The Match Rating Approach was considered because it is designed for indexing names.

The results of this test can be seen in Figure 1. The Double Metaphone algorithm is the best performing, and so, was selected for integration with the modified GED, the results of which can be seen in Table 7.

Accuracy	0.596
Time/s	13

Table 7: Results of reduced modified Double Metaphone

6.2 Analysis

This algorithm is slightly worse, achieving 0.002 lower accuracy, but also runs in 2.3% of the time. The results for the names compared for previous systems are fairly similar, as seen in Table 8. Table 9 shows a few names that the GED and Double Metaphone predicted differently. The Double Metaphone was far more successful at more esoteric names, whereas GED was more successful in common English names

Persian	**Prediction**	**Correct**
RNZLND	renzland	✓
GRNANDR	grenander	✓
BSAK	bosak	✓
NYYVKAMB	newcomb	✓
VANDNBRQ	vandenburgh	✓
KAVDLYNG	quodling	✓
GRASR	graser	grosser
BYSMARK	bismark	bismarck
BRNDYS	brandies	brandeis
DYLYNGR	dillinger	dellinger

Table 8: Examples of reduced Double Metaphone

Persian	**GED**	**Double Metaphone**
PARSVNZ	parsons ✓	persephones
MKGVVAN	macgowan ✓	mcgavin
ANDSVRT	vandivort	endesworth ✓
ZYVTRVD	silverwood	zoeterwoude ✓

Table 9: Examples of Differences Between GED and Double Metaphone

7 Full Dataset

The final test of each system is the full dataset. Results can be seen in Table 10 and Table 11, and are much the same as with the previous sections, with GED having an accuracy notably 0.034 higher. The trade-offs between the 3 systems are evident: Double Metaphone completes almost 50x faster, 4.5 minutes compared to 3.5 hours, at the cost of reduced accuracy. The unreduced GED has an accuracy 10 percentage points higher, at the cost of reduced precision and up to 46 predictions for one Persian name.

System	**Acc.**	**Time/s**
Basic GED		
Basic Modified GED		
Modified GED	0.730	12836
Reduced Modified GED	0.638	12837
Double Metaphone + GED	0.593	263

Table 10: Results of modified GED on Full Dataset

Precision	0.622
Avg. Predictions	1.69
Max. Predictions	46

Table 11: Extra metrics of modified GED on Full Dataset

8 Conclusions

Overall, through careful analysis of the effects of transliteration and transcoding, as well as consideration of the context of edit distance and adjusting parameters accordingly, the final reduced GED achieved an accuracy 0.344, or 120%, higher than the Basic GED. Th next step to improve the accuracy would be to change reduction step of the system, either by further editing the n-gram metric, or by finding a more effective alternative.

References

Chris Little. 2014-2015. abydos. https://github.com/chrislit/abydos.

Daniel Lindsley. 2014. pylev. http://github.com/toastdriven/pylev.

Graham Poulter. 2012. abydos. http://github.com/gpoulter/python-ngram.

Hiroyuki Tanaka. 2013. editdistance. https://www.github.com/aflc/editdistance.

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

Sarvnaz Karimi, Andrew Turpin, and Falk Scholer. 2007. Corpus effects on the evaluation of automated transliteration system. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL07)*, pages 640–647.