The Amirkabir Machine Transliteration System for NEWS 2011: Farsi-to-English Task

Najmeh Mousavi Nejad
Department of Engineering,
Islamic Azad University,
Science & Research
Branch, Punak, Ashrafi
Isfahani , Tehran, Iran
najme.mousavi@gmail.com

Shahram Khadivi
Department of Computer
Engineering, Amirkabir
University of Technology
424 Hafez Ave, Tehran,
Iran 15875-4413
khadivi@aut.ac.ir

Kaveh Taghipour
Department of Computer
Engineering, Amirkabir
University of Technology
424 Hafez Ave, Tehran,
Iran 15875-4413
k.taghipour@aut.ac.ir

Abstract

In this paper we describe the statistical machine transliteration system of Amirkabir University of Technology, developed for NEWS 2011 shared task. This year we participated in English to Persian language pair. We use three systems for transliteration: the first system is a maximum entropy model with a new proposed alignment algorithm. The second system is Sequitur g2p tool, an open source grapheme to phoneme convertor. The third system is Moses, a phrased based statistical machine translation system. In addition, several new features are introduced to enhance the overall accuracy in the maximum entropy model. The results show that the combination of our maximum entropy system with Sequitur g2p tool and Moses lead to a considerable improvement over each system result.

1 Introduction

This paper describes the statistical machine transliteration system used for participation in the NEWS 2011 shared task workshop. We participated in English to Persian task and used three different systems for transliteration generation.

There have been a few researches on Persian language (Karimi et al., 2007). The quality of transliterated names has been improved in the past studies. However, the proposed method is language specific and the algorithm is designed for Persian language. We present two combined transliteration systems. The first system is a combination of a maximum entropy model along with our proposed alignment algorithm and Sequitur g2p tool. The second system is a combination of our maximum entropy system and Moses. Our training and test data is English to Persian set from NEWS 2011 Name Transliteration Shared Task (Zhang et al., 2011). We use openNlP maximum entropy package to train our system. We define new features for discriminative training. Moreover a new approach for aligning name pairs is proposed.

2 The Transliteration Process

Our Maximum Entropy transliteration system has the following steps:

1. Preprocessing
2. Alignment of name pairs
3. Definition of proper features for aligned names
4. Training the model to produce features weight

2.1 Preprocessing

Preprocessing plays an important role in many NLP Applications. The amount and kind of processing done depends on the nature of the language. Since there are some letters in Persian language which have more than one Unicode (for example “ی”), we run a normalization tool on the training set to uniform the letters.

2.2 Alignment of Name Pairs

The features for maximum entropy training are extracted from aligned names. Our proposed alignment method is a two-dimensional Cartesian coordinate system. The horizontal axis is labeled with the source name and the vertical axis is labeled with the target name (or vice versa). A line is drawn from the coordinate (0,0) to the point with coordinate (source_name_length , target_name_length). We mark the

Proceedings of the 2011 Named Entities Workshop, IJCNLP 2011, pages 91–95,
Chiang Mai, Thailand, November 12, 2011.

corresponding cell in each column of the alignment matrix which has the less distance to the line. A single line is not enough for a name pair and is only suitable for names with equal length. For more complex alignments, some fixed points are needed in order to draw the lines. In Figure 1, (bb,ب) and (n,ن) are fixed points and the following alignments are achieved:

(g,گ) , (i,ی) , (bb,ب) , (o,و) , (n,ن) , (i,ی)

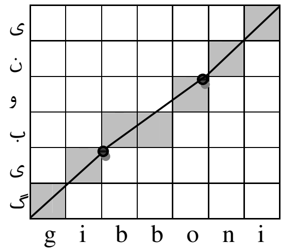

Figure 1.Alignment matrix of (gibboni,گیبونی)

Based on the fact that our goal is to design a language independent transliteration system, an automatic way to find the fixed points is of interest. We introduce FPA algorithm (Fixed Points Alignment) which is an unsupervised approach that adopts the concept of EM training. In the expectation step the training name pairs are aligned using the current model and in the maximization step the most probable alignments are added to the fixed point set. A brief sketch of FPA algorithm is presented in Figure 2. Line 5 to 11 shows the process of updating the fixed points set. In line 7 forcedAlignment means using the current ME model to transliterate source name with the condition that the produced transliterations should be the same as the target name. This condition guarantees the convergence of the algorithm. Line 9 is the last step in producing fixed point set. |k| is the number of distinct segments in the best path set and $p(\tilde{T}_k|\tilde{S}_k)$ is the probability of the $\tilde{T}_k|\tilde{S}_k$ transformation rule. Once the probabilities are calculated, they are compared to a predefined threshold (in our case threshold is 0.9).

2.3 Definition of Proper Features for Aligned Names

We define two types of features: consonant-vowel and n-gram. For both types current context (letter), two past and two future contexts are used. We choose a window with a size of 5, since lower or higher length would have degraded the results.

2.3.1 Consonant-Vowel Features

Every language has a set of consonant and vowel letters. The consonant letters can be divided into different groups based on their types (Table 1).

Plosive (stop)	p , b , t , d , k , g , q
Fricative	f , v , s , z , x , h
Plosive-Fricative	j , c
Flap (tap)	r
Nasal	m , n
Lateral approximant	l , y

Table 1. Six group of consonants

Most combinations of consonant-vowel features were tested for English to Persian transliteration. We have found the following consonant-vowel features are the most effective ones for generating current target letter (t_n). S_i is used to represent the source name characters and t_i represents the target name characters. CV is an abbreviation for consonant- vowel. Note that the consonant letters are divided according to Table $CV_{s_{n-2}}, CV_{s_{n-1}}, CV_{s_n}, CV_{s_{n+1}}, CV_{s_{n+2}}, CV_{t_{n-1}}$. The consonant-vowel features improve transliteration, but still are not sufficient. Therefore we need n-gram features.

```
1:  while( fixedPoints != oldFixedPoints) {
2:    oldFixedPoints = fixedPoints;
3:    fixedPoints = updateFixedPoints(whole_training_corpus)
4:  }
5:  Function updateFixedPoints(training_data){
6:    for( all name pairs) do
7:        A = forcedAlignment(sourceName, targetName, currentModel)
8:    for (all segment pairs in A) do
9:        $p(\tilde{T}_k|\tilde{S}_k) = \frac{\sum p(\tilde{T}_k,\tilde{S}_k)}{\sum_{\acute{T}} p(\acute{T},\tilde{S}_k)}$ , $p(\tilde{S}_k|\tilde{T}_k) = \frac{\sum p(\tilde{T}_k,\tilde{S}_k)}{\sum_{\acute{S}} p(\acute{S},\tilde{T}_k)}$
10:   if (p > threshold) {  add transformation rule to the fixedPoints}
11: }
```

Figure 2. Sketch of the FPA algorithm

2.3.2 N-gram Features

In n-gram features for source name, two past and two future contexts are used (a window with a size of 5). For the target name however, only two past contexts are used (since we don't have future context yet).

Using S to demonstrate the source name and T to demonstrate the target name, the n-gram features for each name can be summarized as:

$s_{n-2} s_{n-1}\ s_n\ s_{n+1}\ s_{n+2}$
$t_{n-2} t_{n-1}\ \times\ \times\ \times$

For any language pair, all combinations of s_i and t_i can be used to define a feature. We tested almost any combination of above features for English to Persian transliteration. The results show that t_{n-2} does not help in better transliteration. Because written Persian omits short vowels, and only long vowels appear in texts. So t_{n-2} is completely irrelevant for generating current Persian letter. But other contexts lead to a better transliteration.

The details of FPA algorithm and feature selection strategies are explained in our research paper which was accepted by NEWS 2011.

2.3 Training the Model and Producing Features Weight

As mentioned earlier, we use openNlP maximum entropy package in the training stage. The features which were extracted in the previous section are inputs for maximum entropy model. After a number of iterations, ME builds the model and produces the features weight. These weights will be used in the test stage.

Some names in the workshop dataset have more than one transliteration. Several experiments were done to study the effect of multi transliteration dataset on our system. Table 2 shows the results. The numbers and phases in the table are defined as follows:

Phase 1: updating the fixed points set

Phase 2: finding features weight

Approach 1: each Persian variant and corresponding English name is considered as one name pair. So if a line in the training file has one English name and 5 Persian transliterations, we will have 5 name pairs for that line. This approach causes many similar alignments to be added to the feature file for a single line in the training file.

Approach 2: This approach is similar to approach 1, except that we add distinct alignments to the feature file for each line in the training set. In other words all alignments of the first Persian transliteration are added to the feature file. For other variants only the alignments which were not seen in the previous Persian transliterations, are added to the file.

Approach 3: we assign an equal weight to each Persian transliteration of an English name. For example if an English name has 4 Persian transliteration, the value of each name weight will be 0.25.

Approach 4: only one Persian name is selected for training. The selection process uses the previous model to estimate the best Persian transliteration.

The best word accuracy in Table 2 belongs to the last row. So in the rest of the paper we use approach 2 for the first phase and approach 1 for the second phase.

Phase 1	Phase 2	WA	CA
Approach 1	Approach 2	65.7	82.4
Approach 1	Approach 1	66.8	82.5
Approach 3	Approach 1	66.8	82.5
Approach 4	Approach 2	67.2	82.7
Approach 2	Approach 2	67.3	82.7
Approach 4	Approach 1	68.2	82.9
Approach 2	Approach 1	68.3	82.9

Table 2. The Effect of multi transliteration dataset on word accuracy and character accuracy in Top-1 tested on the development set

3 System Combination

System combination is the method of combining stand alone systems to achieve a better result. We have three separate systems for transliteration which generate a reasonable output. The first System is the ME model along with our new alignment approach. The second system is the open source Sequitur G2P which is a grapheme to phoneme conversion tool (Bisani and Ney, 2010). Considering the transliteration direction, the names in the source language are regarded as graphemes and the names in the target language as phonemes. The third System is Moses, a phrased based statistical machine translation system. In order to have an accurate transliteration system with a phrase-based statistical translation model, Moses is trained with an unconstrained phrase length. Having no limit for the maximum phrase length is feasible in the transliteration case since the number of phrase pairs are much less when compared to the translation. Having no restriction for the phrase length enables the model to learn all proper phrases and also to perform as a translation memory. In addition, the decoder is not permitted to reorder the phrases by setting the distortion limit to zero. Moreover, the beam threshold, hypothesis stack size and the translation table limit is set to have maximum performance.

The final combined system should produce10 candidates for each name in the test data. To achieve this goal, the first combined system which is a combination of Sequitur g2p and MEM with FPA, has the following steps: First g2p produces 50 candidates for each name, ranked by the probability that the model assigns to them (P_1). Therefore if the number of test names are N, we will have N*50 name pairs. Then we apply forceAlignmnet to each pair which was described in Section 2.2. This process produces another probability for each pair (P_2), which is the multiplication of the best path edges in the search tree (see Figure 2 for further details). Now we can use a linear combination of P_1 and P_2. The final probability for each pair is:

$$P_{final} = \lambda * P_1 + (1 - \lambda) * P_2 \quad (3.1)$$

Once λ is found, 10 best transliterations which have highest P_{final}, are enumerated as final transliterations.

The second combined system is a combination of Moses and MEM with FPA. The process is similar to the first combined system. The difference is the value of λ. The values of λ for each combined system are reported in the next section.

4 Results

We report our results on the development data provided by the NEWS 2011 task. For the development runs, we use the training set for training and the development set for testing. The best combinations of features, founded in section 2.3, are included in the training stage.

We split development data into two half. The first half is used for tuning λ and the second half is used for systems evaluation. Table 3 shows word accuracy in Top-1 and MRR in Top-10 for the five systems. The value of λ for the forth system is set to 0.57 and for the fifth system is set to 0.7.

The workshop released train and development dataset have overlap and some names in the training set are repeated in the development set. Therefore a memory based approach will improve the results very much. In this approach if the test data is observed in the training set, its transliterations are put on top of the N-best list. The accuracy in Top-1 with memory based approach for the forth system is 86.4 and for the fifth system is 86.0.

ID	Systems	WA	MRR	F-Score	MAP$_{ref}$
1	MEM with FPA	66.5	77.5	94.6	65.5
2	Sequitur G2P	67.7	79.5	95.0	66.9
3	Moses	67.5	78.8	93.8	66.5
4	1 combined with 2	70.0	81.0	95.2	69.2
5	1 combined with 3	68.2	79.7	94.9	67.1

Table 3. Results on the second half of the development set (in %)

5 Conclusions

In this paper, we presented a language-independent alignment method for transliteration. Discriminative training is used in our system and numbers of new features are defined in the training stage. Furthermore a new grapheme to phoneme tool is recommended for transliteration task, assuming one side as graphemes and the other side as phonemes. Additionally, a phrase-based statistical translation model is configured to have maximum transliteration accuracy and is used as one of the independent components of the system combination process. Results showed that the combination of three systems improves overall accuracy.

Acknowledgments

This work has been partially supported by Iranian Research Institute for ICT (Ex. ITRC) under grant 500/1141. The authors wish to thank the anonymous reviewers for their criticisms and suggestions.

References

Bisani, M., Ney, H., Joint-Sequence Models for Grapheme-to-Phoneme Conversion, Speech Communication (2008), doi: 10.1016/j.specom.2008.01.002

Fraser, A., Marcu, D., Semi-Supervised Training for Statistical Word Alignment, Proceedings of ACL-2006, pp. 769-776, Sydney, Australia

Goto, I., Kato, N., Uratani, N., Ehara, T., Transliteration Considering Context Information Based on the Maximum Entropy Method, In Proc. Of IXth MT Summit. (2003)

Jiampojamarm, S., Kondrak, G., Letter-Phoneme Alignment: An Exploration, Proceedings of the 48th Annual Meet-ing of the Association for Computational Li-nguistics, pages 780–788, Uppsala, Sweden, 11-16 July 2010.

Josef, F., Ney. H., Discriminative Training and Maximum Entropy Models for Statistical Machine Translation, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 295-302.

Karimi, S., Scholer, F., Turpin, A., Collapsed Consonant and Vowel Models: New Approaches for English-Persian Transliteration and Back-Transliteration, The 45th Annual Meeting of the Association for Computational Linguistics (ACL'07), pages 648-655, Prague, Czech Republic, June 2007.

Karimi, S., Machine Transliteration of Proper Names between English and Persian, A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy, BEng. (Hons.), MSc.

Koehn, P., Hoang, H., Birch A., CallisonBurch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer C., Bojar, O., Constantin, A., Herbst E., 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07) – Companion Volume, June.

OpenNLP Maximum EntropyPackage Available at http://incubator.apache.org/opennlp/

Yoon, S., Kim, K., Sproat, R., "Multilingual Transliteration Using Feature based Phonetic Method", Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 112–119, Prague, Czech Republic, June 2007, Association for Computational Linguistics.

Zelenko, D., Aone. C., Discriminative Methods for Transliteration, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 612–617, Sydney, July 2006.

Zhang, M., Kumaran, A. , Li, H., Whitepaper of NEWS 2011 Shared Task on Machine Transliteration, In Proceedings of the ACL-IJCNLP 2011 Named Entity Workshop