Mining Transliterations from Wikipedia using Pair HMMs

Peter Nabende
Alfa-Informatica, University of Groningen
The Netherlands
p.nabende@rug.nl

Abstract

This paper describes the use of a pair Hidden Markov Model (pair HMM) system in mining transliteration pairs from noisy Wikipedia data. A pair HMM variant that uses nine transition parameters, and emission parameters associated with single character mappings between source and target language alphabets is identified and used in estimating transliteration similarity. The system resulted in a *precision* of 78% and *recall* of 83% when evaluated on a random selection of English-Russian Wikipedia topics.

1 Introduction

The transliteration mining task as defined in the NEWS 2010 White paper (Kumaran et al., 2010) required identifying single word transliteration pairs from a set of candidate transliteration pairs. In the case of Wikipedia data, we have a collection of corresponding source and target language topics that can be used for extracting candidate transliterations. We apply a pair HMM edit-distance based method to obtain transliteration similarity estimates. The similarity estimates for a given set of source and target language words are then compared with the aim of identifying potential transliteration pairs. Generally, the pair HMM method uses the notion of transforming a source string to a target string through a series of edit operations. The three edit operations that we consider for use in transliteration similarity estimation include: *substitution*, *insertion*, and *deletion*. These edit operations are represented as hidden states of a pair HMM. Depending on the source and target language alphabets, it is possible to design or use a specific pair HMM algorithm for estimating paired character emission parameters in the edit operation states, and transition parameters for a given design of transitions between the pair HMM's states. Before applying the pair HMM method, we use external datasets to identify a pair HMM variant that we consider as suitable for application to transliteration similarity estimation. We then use the shared task datasets to train the selected pair HMM variant, and finally apply an algorithm that is specific to the trained pair HMM for computing transliteration similarity estimates. In section 2, we discuss transliteration similarity estimation with regard to applying the pair HMM method; section 3 describes the experimental setup and results; section 4 concludes the paper with pointers to future work.

2 Transliteration Similarity Estimation using Pair HMMs

To describe the transliteration similarity estimation process, consider examples of corresponding English (as *source* language) and Russian (as *target* language) Wikipedia topics as shown in Table 1. Across languages, Wikipedia topics are written in different ways and all words in a topic could be important for mining transliterations. One main step in the transliteration mining task is to identify a set of words in each topic for consideration as candidate transliterations. As seen in Table 1, it is very likely that some words will not be selected as

id	**English topic**	**Russian topic**
1	Johnston Atoll	Джонстон (атолл)
2	Oleksandr Palyanytya	Паляница, Александр Витальевич
3	Ministers for Foreign Affairs of Luxembourg	Категория:Министры иностранных дел Люксембурга
	...	...

Table 1: Example of corresponding English Russian Wikipedia topics

Proceedings of the 2010 Named Entities Workshop, ACL 2010, pages 76–80,
Uppsala, Sweden, 16 July 2010.

candidate transliterations depending on the criteria for selection. For example, if a criterion is such that we consider only words starting with uppercase characters for English and Russian datasets, then the Russian word 'атолл' in the topic pair 1 in Table 1 will not be used as a candidate transliteration and that in turn makes the system loose the likely pair of 'Atoll, атолл'. After extracting candidate transliterations, the approach we use in this paper takes each candidate word on the source language side and determines a transliteration estimate with each candidate word on the target language side. Consider the example for topic id 1 in Table 1 where we expect to have 'Johnston' and 'Atoll' as candidate source language transliterations, and 'Джонстон' and 'атолл' as candidate target language transliterations. The method used is expected to compare 'Johnston' against 'Джонстон' and 'атолл', and then compare 'Atoll' to the Russian candidate transliterations. We expect the output to be 'Johston, Джонстон' and 'Atoll, атолл' as the most likely single word transliterations from topic pair 1 after sorting out all the four transliteration similarity estimates in this particular case. We employ the pair HMM approach to estimate transliteration similarity for candidate source-target language words.

A pair HMM has an emission state or states that generate two observation sequences instead of one observation sequence as is the case in standard HMMs. Pair HMMs originate from work in Biological sequence analysis (Durbin et al., 1998; Rivas and Eddy, 2001) from which variants were created and successfully applied in cognate identification (Mackay and Kondrak, 2005), Dutch dialect comparison (Wieling et al., 2007), transliteration identification (Nabende et al., 2010), and transliteration generation (Nabende, 2009). As mentioned earlier, we have first, tested two pair HMM variants on manually verified English-Russian datasets which we obtain from the previous shared task on machine transliteration (NEWS 2009) (Kumaran and Kellner, 2007). This preliminary test is aimed at determining the effect of pair HMM parameter changes on the quality of the transliteration similarity estimates. For the first pair HMM variant, no transitions are modeled between edit states; we only use transtion parameters associated with transiting from a start state to each of the edit operation states, and from each of the edit operation states to an end state. The

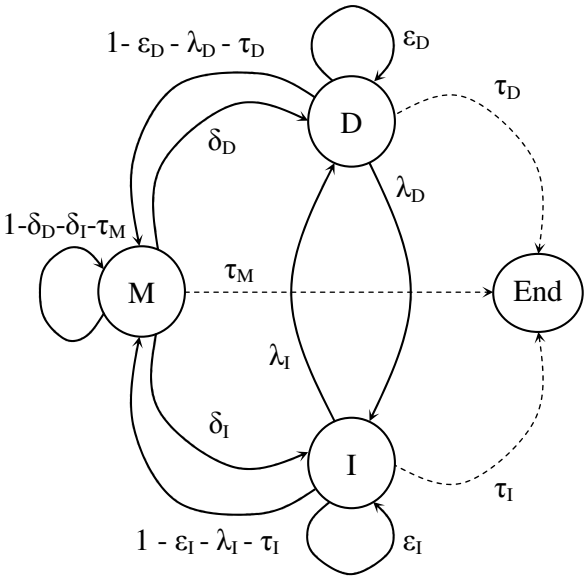

Figure 1: Pair HMM with nine distinct transition parameters. Emission parameters are specified with emitting states and their size is dependent on the characters used in the source and target languages

second pair HMM variant uses nine distinct transition parameters between the pair HMM's states as shown in Figure 1. The node M in Figure 1 represents the substitution state in which emission parameters encode relationships between each of the source and target language characters. D denotes the deletion state where emission parameters specify relationships between source language characters and a target language gap. I denotes the insertion state where emission parameters encode relationships between target language characters and a source language gap. Starting parameters for the pair HMM in Figure 1 are assoicated with transiting from the M state to one of the edit operation states including transiting back to M.

The pair HMM parameters are estimated using the well-known Baum-Welch Expectation Maximization (EM) algorithm (Baum et al., 1970). For each pair HMM variant, the training algorithm starts with a uniform distribution for substitution, deletion, insertion, and transition parameters, and iterates through the data until a local maximum.

A method referred to as *stratified ten fold cross validation* (Olson and Delen, 2008) is used to evaluate the two pair HMM variants. In each fold, 7056 pairs of English-Russian names from the previous shared task on machine transliteration (Ku-

Pair HMM Model		CVA	CVMRR
phmm00edtrans	Viterbi	0.788	0.809
	Forward	0.927	0.954
phmm09edtrans	Viterbi	0.943	0.952
	Forward	**0.987**	**0.991**

Table 2: CVA and CVMRR results two pair HMM variants on a preliminary transliteration identificatio experiment. phmm00edtrans is the pair HMM variant with no transition parameters between the edit states while phmm09edtrans is the pair HMM variant with nine distinct transition parameters.

maran and Kellner, 2007) are used for training and 784 name pairs for testing. The Cross Validation Accuracy (CVA) and Cross Validation Mean Reciprocal Rank (CVMRR) results obtained from applying the Forward and Viterbi algorithms of the two pair HMM variants on this particular dataset are shown in Table 2.

The CVA and CVMRR values in Table 2 suggest that it is necessary to model for transition parameters when using pair HMMs for transliteration similarity estimation. Table 2 also suggests that it is better to use the Forward algorithm for a given pair HMM variant. Based on the results in Table 2, the pair HMM variant illustrated in Figure 1 is chosen for application in estimating transliteration similarity for the mining task.

3 Experimental setup and Results

To simplify the analysis of the source and target strings, the pair HMM system requires unique whole number representations for each character in the source and target language data. This is not suitable for all the different types of writing systems. In this paper, we look at only the English and Russian languages where many characters are associated with a phonemic alphabet and where numbered representations are hardly expected to contribute to errors from loss of information inherent in the original orthography. A preliminary run on Chinese-English[1] datasets from the previous shared task on machine transliteration (NEWS 2009) resulted in an *accuracy* of 0.213 and *MRR* of 0.327 using the pair HMM variant in Figure 1. In the following subsection we discuss some data preprocessing steps on the English-Russian Wikipedia dataset.

3.1 English and Russian candidate transliteration extraction

The English-Russian Wikipedia dataset that was provided for the transliteration mining task is very noisy meaning that it has various types of other entities in addition to words for each language's orthography. A first step in simplifying the transliteration mining process was to remove any unnecessary entities.

We observed the overlap of writing systems in both the English and Russian Wikipedia datasets. We therefore made sure that there is no topic where the same writing system is used in both the English and Russian data. Any strings that contain characters that are not associated with the writing systems for English and Russian were also removed.

We also observed the presence of many temporal and numerical expressions that are not necessary on both the English and Russian Wikipedia datasets. We applied different sets of rules to remove such expressions while leaving any necessary words.

Using knowledge about the initial formatting of strings in both the English and Russian data, a set of rules was applied to split most of the strings based on different characters. For example almost all strings in the English side had the underscore '_' character as a string separator. We also removed characters such as: colons, semi-colons, commas, question marks, exclamation marks, dashes, hyphens, forward and back slashes, mathematical operator symbols, currency symbols, etc. Some strings were also split based on string patterns, for example where different words are joined into one string and it was easy to identify that the uppercase character for each word still remained in the combined string just like when it is alone. We also removed many abbreviations and titles in the datasets that were not necessary for analysis during the transliteration mining process.

After selecting candidate words based on most of the criteria above, we determine all characters in our extracted candidate transliteration data and compare against those in the shared task's seed data (Kumaran et al., 2010) with the aim of finding all characters that are missing in the seed data. Matching transliteration pairs with the the miss-

[1] In this case Chinese is the source language while English is the target language

ing characters are then hand picked from the candidate words dataset and added to the seed data before training the pair HMM variant that is selected from the previous section. The process for identifying missing characters and words that have them is carried out seperately for each language. However, a matching word in the other language is identified to constitute a transliteration pair that can be added to the seed dataset. For the English-Russian dataset, we use 142 transliteration pairs in addition to the 1000 transliteration pairs in the initial seed data. We hence apply the Baum-Welch algorithm for the selected pair HMM specification from section 2 on a total of 1142 transliteration pairs. The algorithm performed 182 iterations before converging for this particular dataset.

3.2 Results

To obtain transliteration similarity measures, we apply the Forward algorithm of the trained pair HMM from section 3.1 to all the remaining Wikipedia topics. For each word in an English topic, the algorithm computes transliteration similarity estimates for all words in the Russian topic. After observing transliteration similarity estimates for a subset of candidate transliteration words, we specify a single threshold value (*th*) and use it for identifying potential transliteration pairs. A threshold value of 1×10^{-13} was chosen after observing that many of the pairs that had a similarity estimate above this threshold were indeed transliteration pairs. Therefore, a pair of words was taken as a potential transliteration pair only when its transliteration estimate (*tr_sim*) was such that $tr\_sim > th$. This resulted in a total of 299389 potential English-Russian transliteration pairs. This collection of potential transliteration pairs has been evaluated using a random set of corresponding English and Russian Wikipedia topics as specified in the NEWS 2010 White paper for the transliteration mining task (Kumaran et al., 2010). Table 3 shows the *precision*, *recall*, and *f-score* results[2] that were obtained after applying the Forward algorithm for the pair HMM of Figure 1.

Despite using the pair HMM method with its basic probabilistic one-to-one mapping for each of the source target character representations, the result in Table 3 suggests a promising application of pair HMMs in mining transliterations from Wikipedia.

Model	*precision*	*recall*	*f-score*
phmm09edtrans	0.780	0.834	0.806

Table 3: Evaluation results for the Pair HMM of Figure 1 on a random selection of 1000 corresponding English Russian Wikipedia topics.

4 Conclusions and Future Work

We have described the application of Pair HMMs to mining transliterations from Wikipedia. The transliteration mining evaluation results suggest a valuable application of Pair HMMs to mining transliterations. Currently, the pair HMM system is considered to be best applicable to languages whose writing system mostly uses a phonemic alphabet. Although an experimental test run was done for Chinese-English data, a conclusion about the general applicability of the pair HMM necessitates additional tests using other language pairs such as Hindi and Tamil which were also part of the shared task.

As future work, we would like to investigate the performance of Pair HMMs on additional writing systems. This may require additional modifications to a pair HMM system to minimize on input formatting errors for other types of writing systems. It is also necessary to determine the transliteration mining performance of pair HMMs when more tolerant criteria are used on the noisy Wikipedia data. Currently, the pair HMM is applied in its most basic form, that is, no complex modifications have been implemented for example modeling for context in source and target language words, and other factors that may affect the quality of a transliteration similarity estimate; it should be interesting to investigate perfromance of complex pair HMM variants in transliteration mining.

Acknowledgments

Research in this paper is funded through a second NPT (Uganda) Project.

References

A Kumaran, Mitesh Khapra, and Haizhou Li. 2010. Whitepaper on NEWS 2010 Shared Task on

[2]The numbers in Table 3 were obtained from a post evaluation after correcting a number of processing errors in the pair HMM transliteration mining system. The errors initially led to relatively lower values associated with the measures in this Table. The values in this Table are therefore not part of the initial shared task results

Transliteration Mining.

A Kumaran and Tobias Kellner. 2007. A Generic Framework for Machine Transliteration. *Proceedings of the* 30^{th} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, pp 721–722, Amsterdam, The Netherlands.

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Annals of Mathematical Statistics*, 41(1):164–171.

David L. Olson and Dursun Delen. 2008. *Advanced Data Mining Techniques*. Springer.

Elena Rivas and Sean R. Eddy. 2001. Noncoding RNA Gene Detection using Comparative Sequence Analysis. *BMC Bioinformatics 2001*, 2:8.

Martijn Wieling, Therese Leinonen, and John Nerbonne. 2007. Inducing Sound Segment Differences using Pair Hidden Markov Models. In John Nerbonne, Mark Ellison, and Grzegorz Kondrak (eds.) *Computing Historical Phonology:* 9^{th} *Meeting of the ACL Special Interest Group for Computational Morphology and Phonology Workshop*, pp 48–56, Prague, Czech Republic.

Peter Nabende. 2009. Transliteration System using Pair HMMs with Weighted FSTs. Proceedings of the Named Entities Workshop, NEWS'09, pp 100–103, Suntec, Singapore.

Peter Nabende, Jorg Tiedemann, and John Nerbonne. 2010. Pair Hidden Markov Model for Named Entity Matching. In Tarek Sobh (ed.) *Innovations and Advances in Computer Sciences and Engineering*, pp 497–502, Springer, Heidelberg.

Richard Durbin, Sean R. Eddy, Anders Krogh, Graeme Mitchison. 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK.

Wesley Mackay and Grzegorz Kondrak. 2005. Computing Word Similarity and Identifying Cognates with Pair Hidden Markov Models. *Proceedings of the ninth Conference on Computational Natural Language Learning (CoNLL 2005)*, pp 40–47, Ann Arbor, Michigan.