Transliteration System using pair HMM with weighted FSTs

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

Abstract

This paper presents a transliteration system based on pair Hidden Markov Model (pair HMM) training and Weighted Finite State Transducer (WFST) techniques. Parameters used by WFSTs for transliteration generation are learned from a pair HMM. Parameters from pair-HMM training on English-Russian data sets are found to give better transliteration quality than parameters trained for WFSTs for corresponding structures. Training a pair HMM on English vowel bigrams and standard bigrams for Cyrillic Romanization, and using a few transformation rules on generated Russian transliterations to test for context improves the system's transliteration quality.

1 Introduction

Machine transliteration is the automatic transformation of a word in a source language to a phonetically equivalent word in a target language that uses a different writing system. Transliteration is important for various Natural Language Processing (NLP) applications including: Cross Lingual Information Retrieval (CLIR), and Machine Translation (MT). This paper introduces a system that utilizes parameters learned for a pair Hidden Markov Model (pair HMM) in a shared transliteration generation task[1]. The pair HMM has been used before (Mackay and Kondrak, 2005; Wieling *et al.*, 2007) for string similarity estimation, and is based on the notion of string Edit Distance (ED). String ED is defined here as the total edit cost incurred in transforming a source language string (S) to a target language string (T) through a sequence of edit operations. The edit operations include: (M)atching an element in S with an element in T; (I)nserting an element into T, and (D)eleting an element in S. Based on all representative symbols used for each of the two languages, emission costs for each of the edit operations and transition parameters can be estimated and used in measuring the similarity between two strings. To generate transliterations using pair HMM parameters, WFST (Graehl, 1997) techniques are adopted. Transliteration training is based mainly on the initial orthographic representation and no explicit phonetic scheme is used. Instead, transliteration quality is tested for different bigram combinations including all English vowel bigram combinations and n-gram combinations specified for Cyrillic Romanization by the US Board on Geographic Names and British Permanent Committee on Geographic Names (BGN/PCGN). However, transliteration parameters can still be estimated for a pair HMM when a particular phonetic representation scheme is used.

The quality of transliterations generated using pair HMM parameters is evaluated against transliterations generated from training WFSTs and transliterations generated using a Phrase-based Statistical Machine Translation (PBSMT) system. Section 2 describes the components of the transliteration system that uses pair HMM parameters; section 3 gives the experimental set up and results associated with the transliterations generated; and section 4 concludes the paper.

2 Machine Transliteration System

The transliteration system comprises of a training and generation components (Figure 1). In the training component, the Baum-Welch Expectation Maximization (EM) algorithm (Baum *et al.*, 1970) is used to learn the parameters of a pair HMM. In the generation component, WFST techniques (Graehl, 1997) model the learned pair HMM parameters for generating transliterations.

2.1 Parameter Estimation for a pair-HMM

A pair HMM has two output observations (Figure 2) that are aligned through the hidden states,

[1] The generation task is part of the NEWS 2009 machine transliteration shared task (Li et al., 2009)

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 100–103,
Suntec, Singapore, 7 August 2009. ©2009 ACL and AFNLP

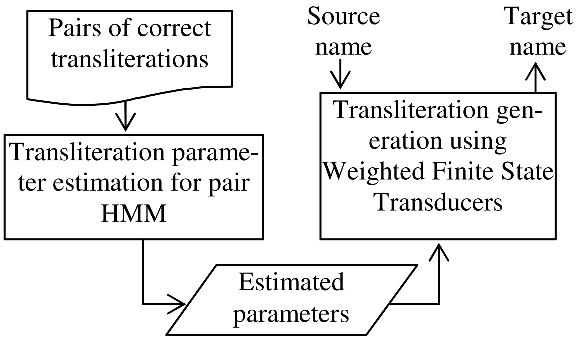
Pairs of correct transliterations
Source name
Target name
Transliteration generation using Weighted Finite State Transducers
Transliteration parameter estimation for pair HMM
Estimated parameters

Figure 1: Machine Transliteration system

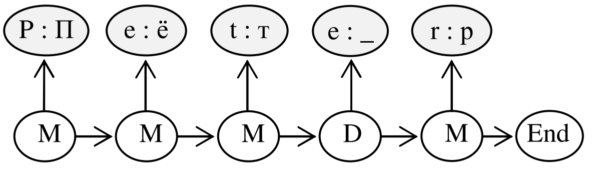

Figure 2: pair-HMM alignment for converting an English string “Peter” to a Russian string “Пётр”

unlike the classic HMMs that have only one observation sequence. The pair HMM structure differs from that of WFSTs in that in WFSTs the input and output symbols and associated weights occur on a transition arc while for the pair HMM, the input and output symbols and associated edit costs are encoded in a node. Two main sets of parameters are learned for the pair HMM: transition parameters (δ, ε, λ, τ_M, τ_{DI}) as shown in Figure 3 for different state transitions; and emission parameters in the (M)atch state and the other two gap states (D and I). s_i in Figure 3 is the i^{th} symbol in the source language string S while t_j is the j^{th} symbol in T.

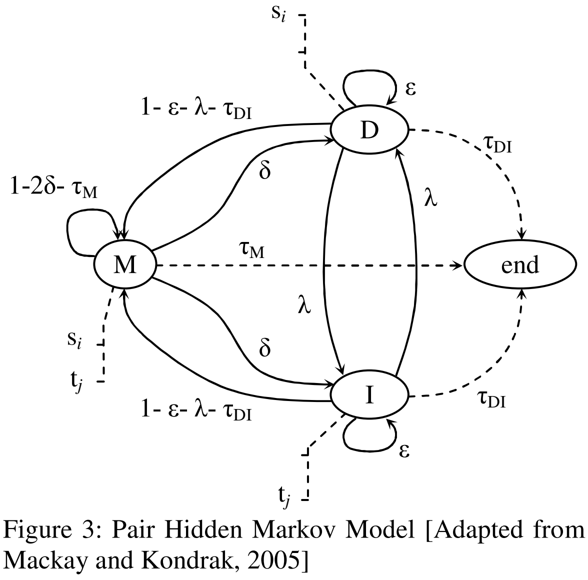

Figure 3: Pair Hidden Markov Model [Adapted from Mackay and Kondrak, 2005]

Pair HMM Emission parameters are stored in matrix form in three tables associated with the edit operations; transition parameters are also stored in matrix form in a table. The emission parameters are $(n \times m)+n+m$ in total; n and m are the numbers of symbols in the pair HMM source language alphabet (V_S) and target language alphabet (V_T) respectively. The parameters of starting in a given edit operation state are derived from the parameters of transiting from the match state (M) to either D or I or back to M.

Although pair HMM training is evaluated against WFST training, there is no major difference in the training approach used in both cases; a forward-backward EM algorithm is used in each case. The main difference is in the structure; for the pair-HMM, the state transition parameter is also incorporated into the weight that measures the level of relationship between the input and output symbol when transformed to a WFST arc.

2.2 Generating Transliterations in WFSTs

A Weighted Finite State Transducer is a finite automaton whose state transitions are labeled with input and output elements and weights that express the level of relationship between the input and output elements. Although the framework of WFSTs has mostly been applied in representing various models for speech recognition (Mohri *et al.*, 2008) including HMMs, WFSTs have as well been used for transliteration (Knight and Graehl, 1998), and are the most suitable for modeling pair HMM constraints for generating transliterations. In the WFST framework, it is possible to specify various configurations associated with constraints inherent in a particular model. Figure 4 shows a WFST that precisely corresponds to the structure of the pair

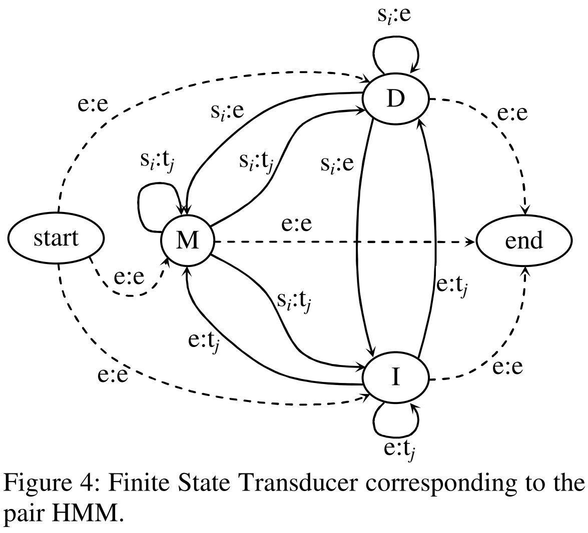

Figure 4: Finite State Transducer corresponding to the pair HMM.

HMM considering the constraints specified for the pair HMM. In Figure 4, e is an empty symbol while s_i and s_j are as defined for the pair HMM in Figure 3. Note that, in Figure 4, a start state is needed to model pair HMM parameter constraints for starting in any of the three edit states. However, it is possible to specify a WFST corresponding to the pair HMM with no start state. Various WFST configurations that do not conform to the bias corresponding to the pair HMM constraints had low transliteration quality and for space limitations, are not reported in this paper.

2.3 Transformation Rules

A look into the transliterations generated using pair HMM parameters on English-Russian development data showed consistent mistransliterations mainly due to lack of contextual modeling in the generated transliterations. For example in all cases where the Russian character л ‘l’ precedes the Russian soft sign ь ‘ ' ’, the Russian soft sign was missing, resulting into a loss of transliteration accuracy. Two examples of mistransliterations that do not include the Russian soft sign ь are: крефелд instead of крефельд ‘krefeld’, and билбао instead of бильбао ‘bilbao’. For such cases, simple transformation rules, such as “л→ль” were defined on the output transliterations in a post processing step. 25 transformation rules were specified for some of the mistransliterations to test the effect of modeling context.

2.4 Transliteration using PSMT system

Transliterations generated using pair HMM parameters and WFSTs are evaluated against those generated from a state of the art Phrase-based Statistical Machine Translation system called Moses. Moses has been used before for machine transliteration (Matthews, 2007) and performed way better than a baseline system that was associated with finding the most frequent mappings between source and target transliteration units in the training data. In the PBSMT system, bilingual phrase-tables are used and several components are combined in a log-linear model (translation models, reverse translation model, word and phrase penalties, language models, distortion parameters, etc.) with weights optimized using minimum error rate training. For machine transliteration: characters are aligned instead of words, phrases refer to character n-grams instead of word n-grams, and language models are defined over character sequences instead of word sequences. A major advantage of the PBSMT system over the pair HMM and a WFST models is that the phrase tables (character n-grams) cover a lot of contextual dependencies found in the data.

3 Experiments

3.1 Data Setup

The data used is divided according to the experimental runs that were specified for the NEWS 2009 shared transliteration task (Li *et al.*, 2009): a standard run and non-standard runs. The standard run involved using the transliteration system described above that uses pair HMM parameters combined with transformation rules. The English-Russian datasets used here were provided for the NEWS 2009 shared transliteration task (Kumaran and Kellner, 2009): 5977 pairs of names for training, 943 pairs for development, and 1000 for testing. For the non-standard runs, an additional English-Russian dataset extracted from the Geonames data dump was merged with the shared transliteration task data above to form 10481 pairs for training and development. For a second set of experiments (Table 2), a different set of test data (1000 pairs) extracted from the Geonames data dump was used. For the system used in the standard run, the training data was preprocessed to include representation of bigrams associated with Cyrillic Romanization and all English vowel bigram combinations.

3.2 Results

Six measures were used for evaluating system transliteration quality. These include (Li *et al.*, 2009): Accuracy (ACC), Fuzziness in Top-1 (Mean F Score), Mean Reciprocal Rank (MRR), Mean Average Precision for reference transliterations (MAP_R), Mean Average Precision in 10 best candidate transliterations (MAP_10), Mean Average Precision for the system (MAP_sys). Table 1 shows the results obtained using only the data sets provided for the shared transliteration task. The system used for the standard run is “phmm_rules” described in section 2 to sub section 2.3. “phmm_basic” is the system in which pair HMM parameters are used for transliteration generation but there is no representation for bigrams as described for the system used in the standard run. Table 2 shows the results obtained when additional data from Geonames data dump was used for training and development. In Table 2, “WFST_basic” and “WFST_rules” are systems associated with training WFSTs for the “phmm_basic” and “phmm_rules” systems

metrics models	ACC	Mean F Score	MRR
phmm_basic	0.293	0.845	0.325
Moses_PSMT	0.509	0.908	0.619
phmm_rules	0.354	0.869	0.394
metrics models	**MAP_R**	**MAP_10**	**MAP_sys**
phmm_basic	0.293	0.099	0.099
Moses_PSMT	0.509	0.282	0.282
phmm_rules	0.354	0.134	0.134

Table 1 Results from data sets for shared transliteration task.

metrics models	ACC	Mean F Score	MRR
phmm_basic	0.341	0.776	0.368
phmm_rules	0.515	0.821	0.571
WFST_basic	0.321	0.768	0.403
WFST_rules	0.466	0.808	0.525
Moses_PSMT	0.612	0.845	0.660
metrics models	**MAP_R**	**MAP_10**	**MAP_sys**
phmm_basic	0.341	0.111	0.111
phmm_rules	0.515	0.174	0.174
WFST_basic	0.321	0.128	0.128
WFST_rules	0.466	0.175	0.175
Moses_PSMT	0.612	0.364	0.364

Table 2 Results from additional Geonames data sets.

respectively. Moses_PSMT is the phrase-based statistical machine translation system. The results in both tables show that the systems using pair HMM parameters perform relatively better than the systems trained on WFSTs but not better than Moses. The low transliteration quality in the pair HMM and WFST systems as compared to Moses can be attributed to lack of modeling contextual dependencies unlike the case in PBSMT.

4 Conclusion

A Transliteration system using pair HMM parameters has been presented. Although its performance is better than that of systems based on only WFSTs, its transliteration quality is lower than the PBSMT system. On seeing that the pair HMM generated consistent mistransliterations, manual specification of a few contextual rules resulted in improved performance. As part of future work, we expect a technique that automatically identifies the mistransliterations would lead to improved transliteration quality. A more general framework, in which we intend to investigate contextual issues in addition to other factors such as position in source and target strings and edit operation memory in transliteration, is that of Dynamic Bayesian Networks (DBNs).

Acknowledgments

Funds associated with this work are from a second NPT Uganda project. I also thank Jörg Tiedemann for helping with experimental runs for the Moses PBSMT system.

References

A. Kumaran and Tobias Kellner. 2007. A Generic Framework for Machine Transliteration. *Proceedings of the 30th SIGIR.*

David Matthews. 2007. *Machine Transliteration of Proper Names*. Master's Thesis. School of Informatics. University of Edinburgh.

Jonathan Graehl. 1997. Carmel Finite-state Toolkit. http://www.isi.edu/licensed-sw/carmel/.

Haizhou Li, A. Kumaran, Min Zhang, Vladimir Pervouchine. 2009. Whitepaper of NEWS 2009 Machine Transliteration Shared Task. *Proceedings of ACL-IJCNLP 2009 Named Entities Workshop* (*NEWS* 2009), Singapore.

Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. *Computational Linguistics*, 24 (4): 599-612, MIT Press Cambridge, MA, USA.

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. *The Annals of Mathematical Statistics*, 41(1):164-171.

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: 9th Meeting of the ACL Special Interest Group for Computational Morphology and Phonology Workshop*, pp. 48-56, Prague.

Mehryar Mohri, Fernando C.N. Pereira, and Michael Riley. 2008. Speech Recognition with Weighte Finite State Transducers. In Larry Rabiner and Fred Juang, editors, *Handbook on Speech Processing and Speech Communication, Part E: Speech Recognition*. Springer-Verlag, Heidelberg, Germany.

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.