Evaluation of Dynamic Bayesian Network models for Entity Name Transliteration

Peter Nabende
Alfa Informatica,
Center for Language and Cognition Groningen,
University of Groningen,
Netherlands
p.nabende@rug.nl

Abstract

This paper proposes an evaluation of DBN models so as to identify DBN configurations that can improve machine transliteration accuracy.

1 Introduction

Machine transliteration is the automatic conversion of a word written in one writing system to another writing system while ensuring that the pronunciation is as close as possible to the original word. For example, using the Cyrillic Translit[1] converter, the entity name "Groningen" in English is converted to "Гронинген" in Russian. Machine Transliteration is important in various cross-language applications including Machine Translation (MT), Cross Language Information Extraction (CLIE) and Cross Language Information Retrieval (CLIR). Based on the units used for transliteration, four models have been proposed for machine transliteration (Oh *et al*., 2006): grapheme-based, phoneme-based, hybrid, and correspondence-based transliteration models. Different types of techniques have been developed by several researchers under these models aimed at improving machine transliteration performance. One framework that has scarcely been evaluated for machine transliteration is that of Dynamic Bayesian Networks (DBNs). DBNs are an extension of Bayesian Networks that are used to model sequential or temporal information. Hidden Markov Models (HMMs) are considered the simplest of DBNs and have been successfully applied in various Natural Language Processing (NLP) applications. Classic HMM-based models have been used for machine transliteration with good transliteration performance. HMMs, however, have restrictions associated with transition and observation parameter independence assumptions that make it difficult to improve machine transliteration performance. More relatively complex DBN models have been exploited before to explore large model spaces for estimating word similarity (Filali and Bilmes, 2005), and have been found to produce better results, although at the expense of computational complexity. DBN models such as those in (Filali and Bilmes, 2005) are used to easily model context and memory issues that are also very important for machine transliteration (Oh and Choi, 2005).

Preliminary results from application of a specific type of DBN models called pair Hidden Markov Models (pair HMMs) (figure 1) on transliteration discovery between English and Russian datasets show promising precision values ranging from 0.80 to 0.86. Currently, we are investigating performance in a transliteration generation task that uses the parameters that have been learned for a pair HMM. The particular pair HMM being investigated has been adapted from previous work on word similarity estimation (Mackay and Kondrak, 2005; Wieling *et al*., 2007). Pair HMMs, however, retain most of the restrictions associated with the classic HMM based models making it difficult to improve performance in transliteration tasks. The next step is to investigate other DBN models such as those introduced in (Filali and Bilmes, 2005) and new DBN models from this research with the aim of distinguishing DBNs that can improve machine transliteration accuracy while being computationally feasible.

2 Transliteration generation problem

There are two types of transliteration that can be used when transliterating between two languages: Forward transliteration where a word in a

[1] The Cyrillic Translit converter is a web-based transliteration utility that can be accessed online at http://translit.cc/

Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning, pages 99–100,
Tilburg, May 2009.

source language is transformed into target language approximations; and backward transliteration, where target language approximations are transformed back to the original source language. In either direction, the transliteration generation task is to take a character string in one language as input and automatically generate a character string in the other language as output. Most of the approaches to automatic transliteration generation involve segmentation of the source string into transliteration units; and associating the source language transliteration units with units in the target language by resolving different combinations of alignments and unit mappings (Haizhou *et al*., 2004). The transliteration units may comprise of a phonetic representation, a Romanized representation, or can be symbols or a combination of symbols in their original writing system.

3 Application of DBN models for machine transliteration

DBNs have several advantages when applied to the task of generating transliterations. One major advantage is that, complex dependencies associated with different factors such as context, memory and position in strings involved in a transliteration process can be captured.

The challenge then, is to specify DBN models that naturally represent the transliteration generation task while addressing some of the factors above. One suitable approach for the transliteration generation problem that is adapted from previous work is based on estimating string edit distance through learned edit costs (Mackay and Kondrak, 2005; Filali and Bilmes, 2005). The edit costs are associated with string edit operations that are used in converting a source language string (S) to a target language string (T). The edit operations specifically include: substitution (M) (replacing a symbol in S with a symbol in T), insertion (I) (matching a symbol in T against a gap in S), and deletion (D) (matching a symbol in S against a gap in T). Figure 1, illustrates these concepts for the case of a pair HMM for an alignment between the English name "Peter" (Roman alphabet) and its Russian counterpart "Пётр" (Cyrillic) through a sequence of edit operations and symbol emissions. As is the case in (Filali and Bilmes, 2005), it is quite natural to construct DBN models representing additional dependencies in the data which are aimed at incorporating more analytical information. Given a

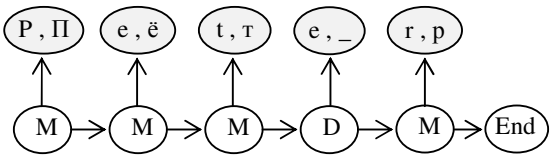

Figure 1: pair-HMM alignment for converting an English string "Peter" to a Russian string "Пётр"

DBN model, inference and learning will involve computing posterior distributions over hidden variables (in the case of transliteration these can be edit operations) given the observed sequences. Fortunately, there exist efficient, generic exact or approximate algorithms that can be adopted for inference and learning a given DBN. By investigating various configurations of DBNs, we hope to provide a more concrete evaluation of applying DBN models for machine transliteration

References

Jong-Hoon Oh, Key-Sun Choi. 2005. An Ensemble of Grapheme and Phoneme for Machine Transliteration. In R. Dale *et al*. (Eds.), *IJCNLP*, *LNAI*, volume 3561: 450-461, Springer-Verlag Berlin Heidelberg.

Jong-Hoon Oh, Key-Sun Choi, Hitoshi Isahara. 2006. A Comparison of Different Machine Transliteration Models. *Journal of Artificial Intelligence Research*, 27 (2006):119-151.

Karim Filali and Jeff Bilmes. 2005. A Dynamic Bayesian Framework to Model Context and Memory in Edit Distance Learning: An Application to Pronunciation Classification. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 338-345, Ann Arbor, June 2005, Association for Computational Linguistics

Li Haizhou, Zhang Min, Su Jian. 2004. A Joint Source-Channel Model for Machine Transliteration. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Article No. 159, Association for Computational Linguistics, Morristown, NJ, USA.

Martijn Wieling, Therese Leinonen, and John Nerbonne. 2007. Inducing Sound Segment Differences using a Pair Hidden Markov Model. In *Proceedings of the 9th Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*, pages 48-56, Prague, Association for Computational Linguistics.

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