

Applying a Dynamic Bayesian Network Framework to Transliteration Identification

Peter Nabende

Alfa-Informatica, University of Groningen
P. O. Box 716, Postcode 9712 EK, Groningen, The Netherlands
E-mail: p.nabende@rug.nl

Abstract

Identification of transliterations is aimed at enriching multilingual lexicons and improving performance in various Natural Language Processing (NLP) applications including Cross Language Information Retrieval (CLIR) and Machine Translation (MT). This paper describes work aimed at using the widely applied graphical models approach of ‘Dynamic Bayesian Networks (DBNs) to transliteration identification. The task of estimating transliteration similarity is not very different from specific identification tasks where DBNs have been successfully applied; it is also possible to adapt DBN models from the other identification domains to the transliteration identification domain. In particular, we investigate the applicability of a DBN framework initially proposed by Filali and Bilmes (2005) to learn edit distance estimation parameters for use in pronunciation classification. The DBN framework enables the specification of a variety of models representing different factors that can affect string similarity estimation. Three DBN models associated with two of the DBN classes originally specified by Filali and Bilmes (2005) have been tested on an experimental set up of Russian-English transliteration identification. Two of the DBN models result in high transliteration identification accuracy and combining the models leads to even much better transliteration identification accuracy.

1. Introduction

Transliteration identification is a task that is aimed at leading to improvements in various NLP applications including: Machine Translation (MT), Cross Language Information Retrieval (CLIR), and automated Question Answering (QA). A common approach to identifying transliterations involves using bilingual or multilingual corpora where the text is represented using different writing systems or alphabets. The transliteration identification task can then be specified as: given a word in one language (a *source* language), identify a matching transliteration or transliterations from another language’s (a *target* language) text. For example given the following name written using English as “Bukenya”, we are interested in determining its most likely target representation out of a set of candidates such as {Набенде ‘Nabende’, Букенья ‘Bukenya’, Фергюсон ‘Ferguson’, ... , Артур ‘Arthur’}. We expect a good model to give the Russian representation “Букенья” as the most likely match for “Bukenya” since it is the correct representation.

Major approaches that have been used in transliteration identification include: rule-based methods (Tsuji et al., 2002), statistical methods (Bilac and Tanaka, 2005; Pouliquen et al., 2006) and various machine learning methods (Lee & Chang, 2003; Kuo et al., 2007; Lee et al., 2006; Udupa, et al., 2008, Udupa et al., 2009; Saravanan and Kumaran, 2008; Oh & Isahara, 2007; Lei et al., 2009; Li et al., 2008; Zhou et al., 2008; Wu et al., 2009). Two main recent categorizations are whether the methods used are generative or discriminative. In this paper, we investigate a class of generative graphical models referred to as ‘Dynamic Bayesian Networks (DBNs)’. DBNs have found successful application in various Identification and Recognition tasks including: Protein Structure

identification (Yao et al., 2008); inferring gene regulatory networks (Shermin & Orgun, 2009); Automatic Speech Recognition (ASR) (Terry & Katsaggelos, 2008), pronunciation classification (Filali and Bilmes, 2005), cognate identification (Kondrak and Sherif, 2006), Human behaviour modelling (Oliver & Horvitz, 2003; Duong et al., 2009), etc. DBNs generalize a variety of models including some of the most common models that have been used in various Natural Language Processing (NLP) tasks such as Hidden Markov Models (HMMs) (Rabiner, 1989). The inference algorithms used in such models can also be seen as instantiations of some of the standard DBN algorithms; for example, the well known forward-backward algorithm used for inference in HMMs can be considered as a specific type of the Message passing algorithm used for inference in Bayesian Networks. Because HMMs are one of the most common graphical models that have previously been applied in identification and recognition tasks, and because of their classification as DBNs, this paper mainly uses them as a reference point while proposing DBN models for application to transliteration identification. Different types of HMMs have already been applied or investigated for transliteration identification (Jeong et al., 1999; Oh & Choi, 2001; Nabende et al., 2010) with impressive transliteration identification results. However, HMMs are restricted by conditional independence assumptions associated with transition and observation parameters that could make it difficult to improve transliteration identification. DBNs allow for a more general representation from which various factorizations can be modelled including learning dependencies between variables that are assumed independent in HMMs (Oliver & Horvitz, 2005). Currently, there exist different approaches that can be used to implement DBN models. This paper is not an attempt to evaluate all the available

approaches but is rather aimed at evaluating one DBN-based edit distance framework that has successfully been applied on tasks that are similar to transliteration identification (Filali and Bilmes, 2005; Kondrak and Sherif, 2006). The DBN-based edit distance framework enables the specification of various models that represent various random variable dependencies that may seem to be important factors in transliteration identification. The paper is organized as follows: section 2 introduces the concept of Dynamic Bayesian Networks and how they can be applied in transliteration identification; section 3 introduces the DBN-based edit distance framework proposed by Filali and Bilmes (2005); section 4 describes the transliteration identification experimental setup and discusses the results obtained from applying the DBN-based framework to transliteration identification; section 5 concludes the paper with pointers to future work.

2. Dynamic Bayesian Networks

DBNs are directed acyclic graphical models that can be used to represent both time-series data that is generated by some causal process and sequence (Natural Language Processing or Biological) data where we are doubtful about the generating mechanism (Murphy, 2002). DBNs generalize HMMs by representing the hidden state as an arbitrary set of random variables with arbitrary conditional independence assumptions (Zweig and Russell, 1998). Murphy (2002) describes a number of advantages that are associated with using DBNs including: the ability to use generic DBN inference and learning procedures that take a lesser time; and the ability to easily change a model once it is represented as a DBN. Figure 1 shows the Bayesian Networks that are needed to completely specify a DBN representation for the standard HMM which has one hidden state variable and one observation variable. Figure 1 shows that when representing a given sequence, we begin from an initial state x_0 that is usually denoted as a start state, and proceed to the state at the next time step, x_1 where we observe y_1 . It

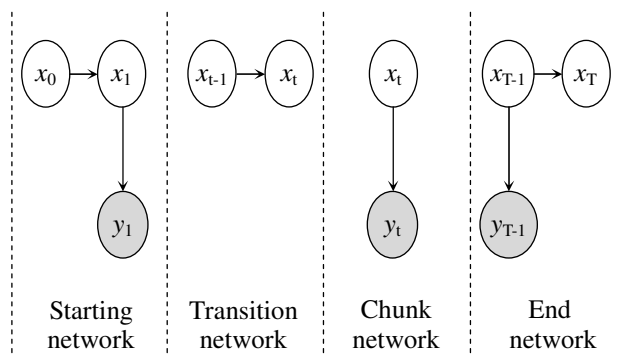


Figure 1: Bayesian networks needed to completely specify a DBN for the simple case of the classic HMMs. Following the most common convention used for representing graphical models (Buntine, 1994), a shaded node represents an observed variable while an unshaded node represents a hidden state variable.

should be noted that the subscripts do not mean that we always proceed to a different state at each time step, instead, they show the movement from one time step to the next, and there is a possibility of staying in a particular state at the next time step. The hidden state variable x_t can have only one value where we realize different observations at different time steps, or it can have different values in which we realize different observations. Figure 1 also shows that a state transition network and a chunk network are needed and they can be *unrolled* either at once or in intervals to enable doing inference over a particular observation sequence. We also need an end network where we stop at a particular end state at time T in the end state x_T . T is the total length of the *unrolled* sequence.

In general, a DBN is defined as a pair $\langle \mathbf{B}_0, \mathbf{B}_\rightarrow \rangle$ where \mathbf{B}_0 is a Bayesian network over an initial distribution over states, and \mathbf{B}_\rightarrow is a two slice Temporal Bayes Net (2-TBN). In the example of Figure 1, \mathbf{B}_0 is equivalent to the starting network while \mathbf{B}_\rightarrow would be composed from the transition and chunk networks. Given a desired window length, an *inter-slice* (e.g. the transition network) topology and *intra-slice* (e.g. the chunk network) topology are used to compose the initial Bayesian network \mathbf{B}_0 and the transition network \mathbf{B}_\rightarrow into a Bayesian network over all the variables within that window (Lerner, 2002) so as to fit a particular observation sequence. Once a particular DBN topology is determined, the remaining task then is how to do inference with it. There are two major categories of inference associated with DBNs: Exact Inference and Approximate inference. In exact inference, a full summation (or integration) over discrete (or continuous) variables is done and it is NP-hard. Depending on the task, exact inference may be intractable. A variety of approximate inference algorithms have been proposed to help overcome the limitations associated with exact inference algorithms. Details about some of the common DBN inference algorithms can be found in (Murphy, 2002).

Currently, there exist a variety of open source and commercial tools that implement the semantics¹ of DBNs. Two commonly used and well implemented tools that are associated with DBNs and are of major interest in this paper are: the Bayes Net Toolbox (BNT) by Murphy (2002), and the Graphical Models Tool Kit (GMTK) by Bilmes and Zweig (1998). A number of the tools differ in the way they represent DBNs and in the algorithms they implement to do inference with the DBNs. For a more detailed comparison of some of the common Graphical Model and Bayesian Network software, please see (Korb & Nicholson, 2004). This paper is mostly concerned with the application of the GMTK toolkit to transliteration identification. In that regard, we adapt a DBN-based edit distance framework implemented using the GMTK toolkit to estimate transliteration similarity. The DBN-based edit distance framework was initially

¹ This means that the graphs are directed and conditional independence relationships are determined by the notion of “d-separation” (Bilmes and Zweig, 2002)

proposed and used in the task of pronunciation classification by Filali and Bilmes (2005) and in cognate identification by Kondrak and Sherif (2006). In the next section we review the features of the GMTK toolkit and introduce the DBN-based edit distance framework.

3. Using GMTK in Transliteration Similarity Estimation

To enable the comparison of various target strings for a given source string, we expect the method used to discriminate among the target strings to determine the most likely target match for the source string. We follow the approach of training a model on correct source target string matches, and thereafter apply the model in estimating the similarity between a source string and a target string. In this section we describe the features of the GMTK toolkit that is used in specifying, training, and testing the DBN models described in this paper

3.1 GMTK Features

GMTK is aimed at implementing various approaches in the Graphical Models framework. GMTK mainly supports the semantics of Bayesian networks (Bilmes and Zweig, 1998). For the transliteration identification task, GMTK has a number of features that support the creation and application of various DBN models:

GMTK generalizes the ability of specifying DBNs through what are referred to as Graphical Model (GM) templates. The templates are defined using GMTK's structure specification language, GMKTL. A template defines a collection of frames² and a chunk specifier. With regard to DBNs, a frame represents a time-slice that contains a set of variables. A frame also includes attributes for each of the variables such as a variable's parents, type of variable (discrete, continuous), parameters associated with a variable and implementation of the parameters (for example discrete probability tables, deterministic relationship, decision trees, etc.) (Bilmes, 2002). The chunk specifier defines two integers which divide the DBN template into a prologue (the first N frames); a repeating chunk; and an epilogue. The chunk frame can be *unrolled* until the network is long enough to represent a specific observation sequence. The GMTK version that we use to estimate the similarity between transliterations supports the creation of junction trees³ that allow for easy implementation of exact inference. The particular algorithm used is referred to as the *Frontier algorithm* and it uses all the hidden nodes in a slice to *d-separate* the past from the future. A number of improvements (Murphy, 2002; Tian and Lu, 2004) have

been suggested over the *Frontier algorithm* and it is necessary to determine the possibility of benefiting from applying other algorithms to the task of transliteration identification. All that is required to use GMTK is to represent our transliteration data in a format that can be processed using the algorithms implemented in GMTK. We follow the approach proposed by Filali and Bilmes (2005) to avoid re-implementing the procedures required in applying GMTK on transliteration data. This approach uses the notion of learning edit distance using DBN graphical models framework. This approach is reviewed in the following subsection.

3.2 The DBN-based Edit Distance Framework (Filali and Bilmes, 2005)

Filali and Bilmes (2005) use a *transducer model* proposed by Ristad and Yianilos (1998) as a *baseline* model in their DBN-based edit distance framework. When specified as a DBN model, the transducer model is referred to as a *Memory-less and Context Independent* (MCI) DBN model. Figure 2 is a graphical representation of the template for the MCI DBN model that was specified using GMTKL. In Figure 2, Z denotes the current edit operation

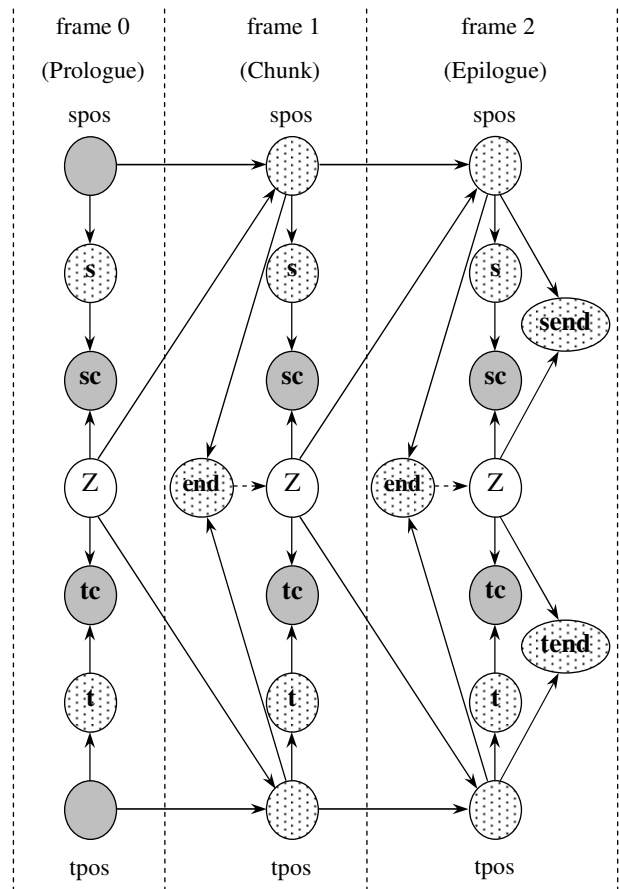


Figure 2: Template for the MCI DBN model. Following the common convention, shaded nodes represent observed variables, while unshaded nodes represent hidden nodes. Nodes with dots represent deterministic hidden variables (Adapted from Filali and Bilmes, 2005).

² The term “frame” is a concept in Automatic Speech Recognition that refers to contiguous, small regions of a speech signal which aid in the identification of phonemes.

³ A junction tree algorithm uses graph theory to form graphs which have the same coupling properties as the original graph but which are ‘easier’ to deal with than the original graph.

(i.e. substitution, deletion, or insertion). $spos$ is a variable representing the current position in the source string, and $tpos$ represents the current position in the target string. s represents the current character in the source string and takes the current position variable in the source string $spos$ as its parent variable, likewise, t represents the current character in the target string and takes the current position variable $tpos$ as its parent variable. The sc and tc nodes are source and target consistency nodes respectively. sc and tc have a fixed observed value 1 and the only configuration of their parents are such that the source component of Z is s or an empty symbol (ϵ) for sc and t or empty symbol (ϵ) for tc , and that Z does not generate empty source and empty target symbols at the same time (Filali and Bilmes, 2005). The end node is a switching parent of Z as illustrated by the dashed arrow and represents the variable that indicates when we are past the end of both source and target strings, i.e when $spos > m$ and $tpos > n$, where m and n are the lengths of the source and target string respectively. The $send$ and $tend$ nodes represent variables that ensure that we are at or past the end of the source and target strings respectively. Filali and Bilmes (2005) implement additional classes of DBN models using the MCI DBN model as a base model. In the other DBN model classes, Filali and Bilmes (2005)

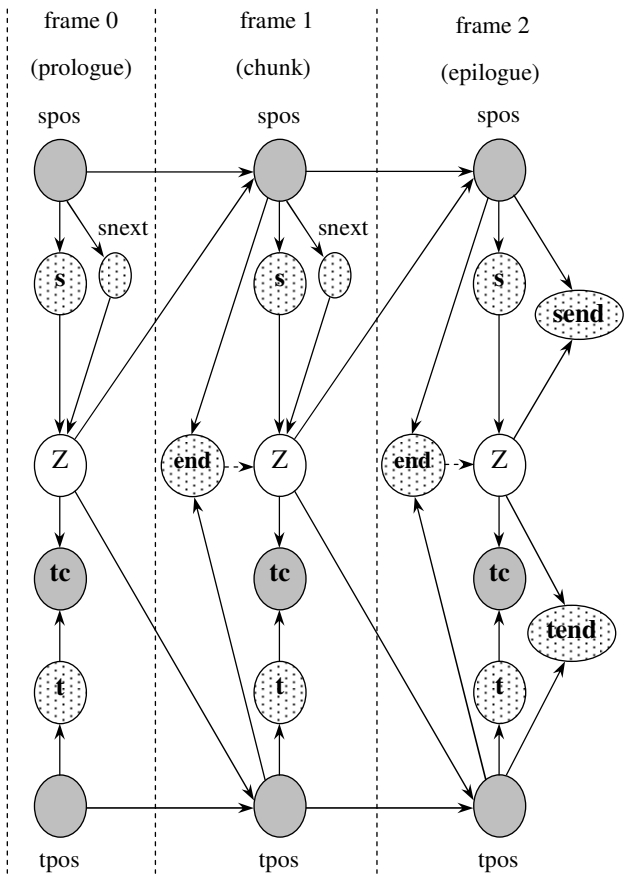


Figure 3: Template for the context dependent DBN model. The consistency variable sc is not represented here since the consistency is enforced through the direct relationship between the edit variable Z and source symbols.

add edges to enable modelling of various factors including: *context* where various context window sizes in a source language string or a target language string or both source and target language strings are represented; *memory* where different dependencies on the hidden edit operation variables are tested; *position* where dependencies on the position in the source or target string are tested; *length* where the length of the sequence is factored into the similarity estimate for the source and target language strings. Figure 3 is a graphical representation of the template for one of the DBN models tested in this paper that is referred to as a *Context dependent (CON) DBN model* where the edit operation Z directly depends on the current and next characters in the source string. Apart from the CON DBN model, the other two DBN models tested in this paper are the *Context dependent length (CON length) DBN model* and the *MCI length DBN model*. Figure 4 shows the template for the *MCI*

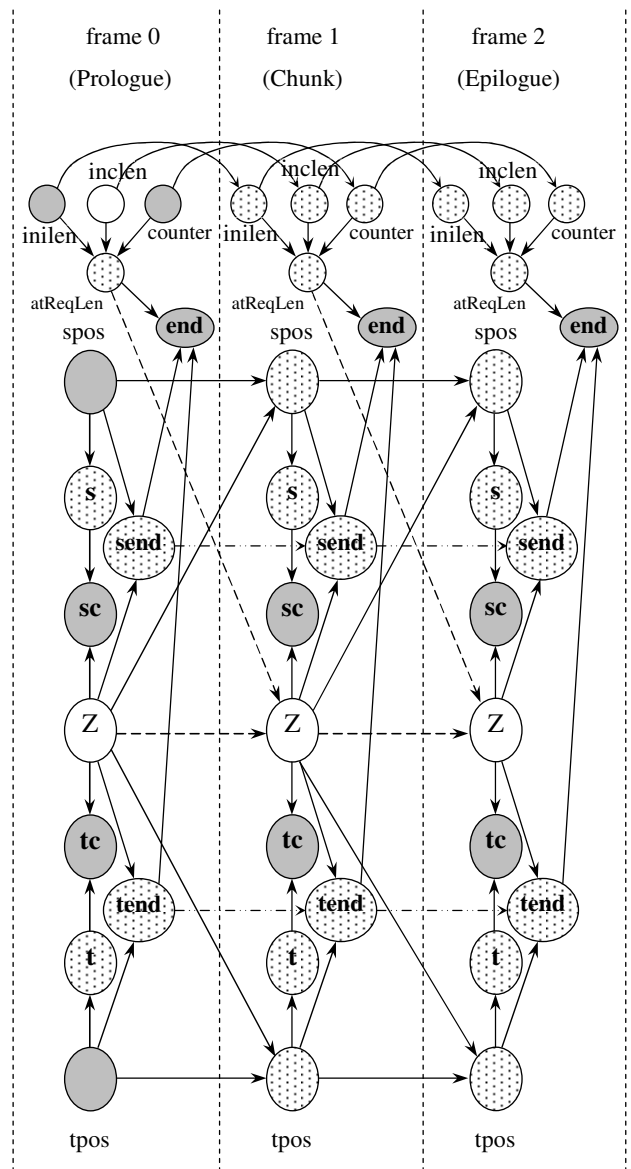


Figure 4: Template for the MCI Length unrolling DBN model. (Adapted from Filali and Bilmes, 2005)

length DBN model. In Figure 4, additional variables are used to implement the logic needed to factor in the length of the edit sequence in the final transliteration similarity estimate. In Figure 4, *inlen*, represents a stochastic hidden random variable whose value added to that of the random variable *inilen* determines the number of allowed edit operations at this point. The variable *counter* is used to determine the frame number and is used to trigger the random variable *atReqLen* when the required frame number is reached. The logic for factoring in the length of the edit sequence in the similarity estimate when using the *CON length* DBN model is specified in a way similar to that of the template in Figure 4. The other difference between the *CON length* DBN model and the *CON* DBN model (see Figure 3), is that the edit operation variable in the *CON length* DBN model has a direct relationship with only the current character in the source string while the edit operation variable in the *CON* DBN model depends on both the current and next characters in the source string. The DBN-based edit distance framework we use, takes as input numeric representations of the source and target strings when training and when determining similarity estimates. The numeric representations seem to simplify the execution of the algorithms that are used in estimating the different types of parameters and for determining similarity estimates. The transformation to numerical representation is, however, only reliable, if the writing systems being analyzed are in such a way that the representation of a symbol as a numeric unit does not lead to loss of information. A good starting point here should therefore be alphabetical writing systems. In logographic writing systems, a symbol represents a complete grammatical word while in a syllabic writing system, a symbol represents or approximates a syllable; a direct transformation of a logogram or a syllabic symbol to a numerical representation as that required in GMTK would lead to loss of information to a very large extent. For the case of logographic and syllabic systems, an intermediate step is required to represent the logograms or syllabic symbols in a much more reasonable format before transforming to the numerical format required when applying GMTK. In this paper, we consider only languages most of whose symbols are part of an alphabetical writing system. More specifically, the DBN models are evaluated on Russian (that uses the Cyrillic alphabet) as the source language and English (that uses the Latin alphabet) as the target language.

4. Experimental Setup and Results

The data used for testing the DBNs described in section 3 is comprised of Russian-English transliteration pairs obtained from the NEWS 2009 shared task on machine transliteration (Kumaran and Kellner, 2007). 7840 Russian-English name pairs are extracted where each name has at least four⁴ characters. The data set is simply

⁴ This constraint is used to enable the evaluation of additional DBN models that were specified by (Filali and Bilmes, 2005). These DBNs are mainly associated with representing

divided in such a way that one tenth⁵ of the dataset comprise testing data while the rest comprise training data. During training, the system of the DBN-based edit distance models take as input a number of correct Russian-English transliteration pairs and iterate through the data while estimating different parameters using a generalized Expectation Maximization (EM) algorithm. Basing on previous work (Filali and Bilmes, 2005; Kondrak and Sherif, 2006), only three iterations are used to train the DBN models since it was reported that more than three iterations led to overfitting of the DBN models. However, it is important to determine the optimal setting for the number of iterations required for training DBN models for transliteration identification. All the DBN models were trained within a single day, and apart from MCI length model, the decoding procedure for the other two DBN models was run over the 784 test names within one day. The difference in execution time is attributed to the methods of representation of some of the parameters in the DBN models for example the use of dense conditional probability tables in the *MCI length* DBN model as compared to the use of sparse CPTs (Bilmes and Zweig, 2002) in the *CON length* DBN model. During scoring, a source name written in Russian was compared against 784 candidate target names written in English. The list of target names was then sorted where names with better scores appeared at the top of the list. The rank of the reference target name for each source name was determined and used to compute Accuracy (ACC) and Mean Reciprocal Rank (MRR) associated with applying each of the DBN models. The accuracy and MRR are computed as follows:

$$ACC = \frac{1}{N} \sum_{i=1}^N \{1 \text{ if } \exists r_i : r_i = cd_{i,1}; 0 \text{ otherwise} \},$$

where N is the total number of names that are tested, r_i is a reference transliteration against which the candidate transliteration $cd_{i,1}$ returned at 1st rank is compared with.

$$MRR = \frac{1}{N} \sum_{i=1}^N \frac{1}{R_i},$$

where N is as defined for ACC above and R_i is the rank of the top reference transliteration in the returned sorted list of candidate transliterations. The results obtained from applying the DBN models described in section 3 are shown in Table 1. Table 2 shows the power values (pvalues) associated with determining the statistical significance of the differences in Accuracy and MRR between the different DBN models and combinations of DBN models. Ideally, the closer the pvalue is to 1, the more statistically significant is the difference. In Table 1,

relationships between variables in different time slices such as the Memory models (for dependencies between the hidden edit operation variables)

⁵ The division of the dataset into one tenth for testing is also aimed at carrying out ten-fold cross validation tests for DBN models that will be executed efficiently

DBN Model	ACC	MRR
MCI-L	0.9031	0.9408
CON (s_i, s_{i+1})	0.9796	0.9834
CON-L (s_i)	0.9834	0.9887
CON-L+CON	0.9949	0.9956
CON-L+CON+MCI-L	0.9962	0.9969

Table 1: ACC and MRR of the DBN models on Russian-English transliteration identification. Bold values indicate relatively higher ACC and MRR compared to the least performing model.

DBN Model	ACC pvalue	MRR pvalue
MCI-L vs. CON (s_i, s_{i+1})	1.00	0.99
vs. CON-L (s_i)	1.00	1.00
vs. CON-L+CON	1.00	1.00
vs. CON-L+CON+MCI-L	1.00	1.00
CON (s_i, s_{i+1}) vs. CON-L (s_i)	0.08	0.14
vs. CON-L+CON	0.77	0.66
vs. CON-L+CON+MCI-L	0.85	0.77
CON-L (s_i) vs. CON-L+CON	0.59	0.34
vs. CON-L+CON+MCI-L	0.71	0.48

Table 2: Power values (pvalue) for differences in accuracy and MRR between the different DBN models and combinations of DBN models at a significance level of 0.05. pvalues in bold show a statistically significant difference between the respective models.

MCI-L is the MCI Length DBN model. CON is the Context dependent DBN models where the edit operation Z depends on the current character and the next character in the source language string, and CON-L is the Context dependent length model as defined in section 3. Table 1 shows that the Context dependent length model performs better than the other two DBN models, but only significantly better than the MCI length DBN model as suggested by the power values in Table 2. The MCI length model performs worst despite taking a longer time to execute on the data set as compared to the other two DBN models. Table 1 also shows that a combination of the CON model and the CON-L model results in a much better transliteration identification accuracy and MRR compared to the individual application of the two DBN models. The performance is improved slightly when the MCI length model is combined with the other two models.

5. Conclusion

This paper introduced the use of DBNs in an experimental transliteration identification task. Results show high transliteration identification quality of the DBN models when applied on Russian-English datasets. The results also suggest that more benefit arises out of a combination of the models as compared to individual applications of

the models. This performance, however, comes at a cost because the DBN models take much longer time to execute. A comparison with the best scoring pair HMMs on the same data set (Nabende, 2010) resulted in similar transliteration identification quality, which implies that more investigation is needed regarding the efficiency of the DBN models when applied to transliteration identification.

As future work, we would like to evaluate the DBN framework on more language pairs apart from Russian-English. This means that languages using different writing systems apart from alphabetical writing systems could also be used for evaluating the DBN-based edit distance framework. Secondly, we expect to evaluate the DBN framework on mining transliterations from Web-based resources such as Wikipedia. Thirdly, it should be interesting to determine the applicability of the DBN models on a transliteration generation task where DBN model parameters are used. Fourthly, the models tested in this paper are only a small fraction of the number of DBN models that can be specified for and can be feasibly applied to transliteration identification; it is important that we investigate the effect of additional factors and dependencies using the framework on transliteration identification quality. Finally, GMTK lacks in a variety of procedures used for inference, it is necessary to determine whether improvements can be achieved by using different inference procedures; the BNT toolkit seems to be equipped with different approaches to inference and is therefore a likely tool to start with.

Acknowledgements

Research presented in this paper is sponsored through a second NPT Uganda project. I thank Karim Filali and Jeff Bilmes for providing scripts for the DBN-based edit distance framework and availing for free the binaries for GMTK, I thank John Nerbonne who initially requested for the DBN scripts. I thank Jörg Tiedemann with whom I was able to grasp DBN concepts. Finally, I thank Erik Tjong Kim Sang who always proof read and provided valuable comments concerning work in this paper.

References

- Bilac, S. & Tanaka, H. (2005). Extracting Transliteration Pairs from Comparable Corpora. In *Proceedings of the eleventh Annual Meeting of the Association for Natural Language Processing*.
- Bilmes, J. & Zweig, G. (1998). The Graphical Models Toolkit: An Open Source System for Speech and Time-series Processing. In *Proc. ICASSP*, Orlando.
- Buntine, W.L. (1994). Operations for Learning with Graphical Models. In *Journal of Artificial Intelligence Research*, AI Foundation and Morgan Kaufmann Publishers, 2 (1994): pp. 159--225.
- Duong, T., Phung, D., Bui, H., & Venkatesh, S. (2009). Efficient Duration and Hierarchical Modelling for Human Activity Recognition. In *Artificial Intelligence*, 173 (7-8): pp. 830--856.

- Filali, K. & Bilmes, J. (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 Association for Computational Linguistics (ACL 2005)*, Ann Arbor, Michigan, USA , pp. 338--345.
- Jeong, K.S., Myaeng, S.H., Lee, J.S. & Choi, K-S. (1999). Automatic Identification and Back-Transliteration of Foreign Words for Information Retrieval. In *Information Processing and Management*, 35 (1999): pp. 523--540.
- Klementiev, A. & Roth, D. (2006). Weakly Supervised Named Entity Transliteration and Discovery from Multilingual Comparable Corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Meeting of the ACL*, Sydney, Australia, pp. 817-824.
- Kondrak, G. & Sherif, T. (2006). Evaluation of Several Phonetic Similarity Algorithms on the Task of Cognate Identification. In *Proceedings of the COLING-ACL Workshop on Linguistic Distances*, Sydney, Australia, pp. 43--50.
- Korb, K.B. & Nicholson, A.E. (2004). *Bayesian Artificial Intelligence*. CRC Press, Florida, USA.
- Kumaran, A. & Kellner, T. (2007). A Generic Framework for Machine Transliteration. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, Amsterdam, The Netherlands, Association for Computing Machinery, Inc., pp. 721-722.
- Kuo, J-S., Li, H., & Yang, Y-K. (2007). A Phonetic Similarity Model for Automatic Extraction of Transliteration Pairs. In *ACM Trans. Asian Language Information Processing*, 6(2) : article 6.
- Lee, C-J. & Chang, J.S. (2003). Acquisition of English-Chinese Transliterated Word Pairs from Parallel Aligned Texts using a Statistical Machine Transliteration Model. In Mihalcea, R. & Pederson, T. (eds.) *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pp. 96--103.
- Lee, C-J., Chang, J.S. & Jang, J-S. R. (2006). Extraction of Transliteration Pairs from Parallel Corpora using a Statistical Transliteration Model. In *International Journal of Information Science*, Elsevier Science Inc., 176 (1) : pp. 67--90.
- Lei., G., Mei-ling, Z., Jian-Min, Y., & Qiao-Ming, Z. (2009). A Supervised Method for Transliterated Person Name Identification. In *Proceedings of the Second International Symposium on Electronic Commerce and Security*, vol. 2, pp. 117--120.
- Lerner, U.N. (2002). *Hybrid Bayesian Networks for Reasoning about Complex Systems*. Ph.D. Thesis, Stanford University.
- Li, H., Kuo, J-S., Su, J., & Lin, C-L. (2008). Mining Live Transliterations Using Incremental Learning Algorithms. In *International Journal Of Computer Processing of Languages (IJCPOL)*, World Scientific, 21(2): pp. 183--203.
- Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference, and Learning*. Ph.D. dissertation, University of California, Berkeley.
- Nabende, P. (2010). Evaluation of a Dynamic Bayesian Network Framework for Transliteration Similarity Estimation. Presentation at the 20th Meeting of Computational Linguistics in the Netherlands (CLIN), Utrecht, The Netherlands.
- Nabende, P., Tiedemann, J., & Nerbonne, J. (2008). Pair Hidden Markov Model for Named Entity Matching. In Tarek, S. (ed.) *Innovations and Advances in Computer Sciences and Engineering*, Springer , pp. 497--502.
- Oh, J-H. & Choi, K-S. (2001). Automatic Extraction of Transliterated Foreign Words using HMM. In *Proceedings of the International Conference of Computer Processing on Oriental Language (ICCPOL2001)*, 19, pp. 433--438.
- Oh, J-H., & Isahara, H. (2006). Mining the Web for Transliteration Lexicons: Joint-Validation Approach. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 254--261.
- Oliver, N. & Horvitz, E. (2005). A Comparison of HMMs and Dynamic Bayesian Networks for Recognizing Office Activities. In *Ardissono, L., Brna, P., & Mitrovic, A. (Eds.) UM 2005, LNAI*, Springer-Verlag Berlin Heidelberg, 3538: pp. 205-215.
- Pouliquen, B., Steinberger, R., Ignat, C., Temnikova, I., Widiger, A., Zaqhouani, W., & Zizka, J. (2006). Multilingual Person Name Recognition and Transliteration. *CORELA – Cognition, Representation, Language*, Poitiers, France, CERLICO, 3(2): pp. 115--128.
- Rabiner, L.R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, 77 (2): pp. 257--286.
- Ristad, E.S. and Yianilos, P.N. (1998). Learning String Edit Distance. In *Trans. on Pattern Recognition and Machine Intelligence*, 20 (5): pp. 522--532.
- Saravanan, K., & Kumaran, A. (2008). Some Experiments in Mining Named Entity Transliteration Pairs from Comparable Corpora. In *Proceedings of the 2nd International Workshop on "Cross Lingual Information Access" Addressing the Information Need of Multilingual Societies*, pp. 26--33.
- Shermin, A. & Orgun, M.A. (2009). Using Dynamic Bayesian Networks to infer Gene Regulatory Networks from Expression Profiles. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, Honolulu, Hawaii, ACM, pp. 799--803.
- Terry, L.H. & Katsaggelos, A.K. (2008). A Phone-Viseme Dynamic Bayesian Network for Audio-Visual Automatic Speech Recognition. In *Proceedings of 2008 International Conference on Pattern Recognition*, Tampa, Florida.
- Tian, F. & Lu, Y. (2004). A DBN Inference Algorithm using Junction Tree. In *Proceedings of the 5th World*

Congress on Intelligent Control and Automation.

- Tsuji, K., Daille, B. & Kageura, K. (2002). Extracting French-Japanese Word Pairs from Bilingual Corpora based on Transliteration Rules. In *Proceedings of the third LREC Conference*, pp. 499--502.
- Udapa, G., Saravanan, K., Bakalov, A., & Bhole, A. (2009). "They Are Out There, If You Know Where to Look": Mining Transliterations of OOV Query Terms for Cross-Language Information Retrieval. In *Advances in Information Retrieval, LCNS*, Springer Berlin / Heidelberg, 5478/2009: pp. 437--448.
- Udapa, R., Saravanan, K., Kumaran, A., Jagarlamudi, J. (2008). Mining Named Entity Transliteration Equivalents from Comparable Corpora. In *CIKM'08*, Napa Valley, California, USA, pp. 1423--1424.
- Wu, X., Okazaki, N., & Tsujii, J. (2009). Semi-Supervised Lexicon Mining from Parenthetical Expressions in Monolingual Web Pages. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the ACL*, Boulder, Colorado, ACL., pp. 424--432.
- Yao, X-Q., Zhu, H., & She, Z-S. (2008). A Dynamic Bayesian Network Approach to Protein Secondary Structure Prediction. In *BMC-Bioinformatics 2008*, 9, 48.
- Zhou, Y., Huang, F., & Chen, H. (2008). Combining Probability Models and Web Mining Models: A Framework for Proper Name Transliteration. In *Journal of Information Technology and Management*, Springer Netherlands, 9: pp. 91--103
- Zweig, G. & Russell, S. (1998). Speech Recognition with Dynamic Bayesian Networks. In *Proc. AAAI-98*, Madison, Wisconsin: AAAI Press.