English—Hindi Transliteration Using Context-Informed PB-SMT: the DCU System for NEWS 2009

Rejwanul Haque, Sandipan Dandapat, Ankit Kumar Srivastava,
Sudip Kumar Naskar and Andy Way
CNGL, School of Computing
Dublin City University, Dublin 9, Ireland
{rhaque,sdandapat,snaskar,asrivastava,away}@computing.dcu.ie

Abstract

This paper presents English—Hindi transliteration in the NEWS 2009 Machine Transliteration Shared Task adding source context modeling into state-of-the-art log-linear phrase-based statistical machine translation (PB-SMT). Source context features enable us to exploit source similarity in addition to target similarity, as modelled by the language model. We use a memory-based classification framework that enables efficient estimation of these features while avoiding data sparseness problems.We carried out experiments both at character and transliteration unit (TU) level. Position-dependent source context features produce significant improvements in terms of all evaluation metrics.

1 Introduction

Machine Transliteration is of key importance in many cross-lingual natural language processing applications, such as information retrieval, question answering and machine translation (MT). There are numerous ways of performing automatic transliteration, such as noisy channel models (Knight and Graehl, 1998), joint source channel models (Li et al., 2004), decision-tree models (Kang and Choi, 2000) and statistical MT models (Matthews, 2007).

For the shared task, we built our machine transliteration system based on phrase-based statistical MT (PB-SMT) (Koehn et al., 2003) using Moses (Koehn et al., 2007). We adapt PB-SMT models for transliteration by translating characters rather than words as in character-level translation systems (Lepage & Denoual, 2006). However, we go a step further from the basic PB-SMT model by using source-language context features (Stroppa et al., 2007). We also create translation models by constraining the character-level segmentations, i.e. treating a consonant-vowel cluster as one transliteration unit.

The remainder of the paper is organized as follows. In section 2 we give a brief overview of PB-SMT. Section 3 describes how context-informed features are incorporated into state-of-art log-linear PB-SMT. Section 4 includes the results obtained, together with some analysis. Section 5 concludes the paper.

2 Log-Linear PB-SMT

Translation is modelled in PB-SMT as a decision process, in which the translation $e_1^I = e_1 \ldots e_I$ of a source sentence $f_1^J = f_1 \ldots f_J$ is chosen to maximize (1):

$$\operatorname*{argmax}_{I,e_1^I} P(e_1^I \mid f_1^J) = \operatorname*{argmax}_{I,e_1^I} P(f_1^J \mid e_1^I).P(e_1^I) \quad (1)$$

where $P(f_1^J \mid e_1^I)$ and $P(e_1^I)$ denote respectively the translation model and the target language model (Brown et al., 1993). In log-linear phrase-based SMT, the posterior probability $P(e_1^I \mid f_1^J)$ is directly modelled as a (log-linear) combination of features (Och and Ney, 2002), that usually comprise M translational features, and the language model, as in (2):

$$\log P(e_1^I \mid f_1^J) = \sum_{m=1}^{m} \lambda_m h_m(f_1^J, e_1^I, s_1^K) + \lambda_{LM} \log P(e_1^I) \quad (2)$$

where $s_1^K = s_1 \ldots s_k$ denotes a segmentation of the source and target sentences respectively into the sequences of phrases $(\hat{e}_1, \ldots, \hat{e}_k)$ and $(\hat{f}_1, \ldots, \hat{f}_k)$ such that (we set $i_0 = 0$) (3):

$$\forall 1 \le k \le K, \quad s_k = (i_k; b_k, j_k),$$
$$\hat{e}_k = e_{i_{k-1}+1} \ldots e_{i_k},$$
$$\hat{f}_k = f_{b_k} \ldots f_{j_k} \quad (3)$$

The translational features involved depend only on a pair of source/target phrases and do not take into account any context of these phrases. This means that each feature h_m in (2) can be rewritten as in (4):

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 104–107,
Suntec, Singapore, 7 August 2009. ©2009 ACL and AFNLP

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) \quad (4)$$

where $\hat{h}_m$ is a feature that applies to a single phrase-pair. Thus (2) can be rewritten as:

$$\sum_{m=1}^{m} \lambda_m \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) = \sum_{k=1}^{K} \hat{h}(\hat{f}_k, \hat{e}_k, s_k) \quad (5)$$

where, $\hat{h} = \sum_{m=1}^{m} \lambda_m \hat{h}_m$. In this context, the translation process amounts to: (i) choosing a segmentation of the source sentence, (ii) translating each source phrase.

3 Source Context Features in Log-Linear PB-SMT

The context of a source phrase $\hat{f}_k$ is defined as the sequence before and after a focus phrase $\hat{f}_k$ = $f_{i_k} \dots f_{j_k}$. Source context features (Stroppa et al., 2007) include the direct left and right context words (in our case, character/TU instead of word) of length *l* (resp. $f_{i_k-1} \dots f_{i_k-l}$ and $f_{j_k+1} \dots f_{j_k+l}$) of a given focus phrase $\hat{f}_k = f_{i_k} \dots f_{j_k}$. A window of size 2*l*+1 features including the focus phrase is formed. Thus lexical contextual information (CI) can be described as in (6):

$$\text{CI} = \{ f_{i_k-l} \dots f_{i_k-1}, f_{j_k+1} \dots f_{j_k+l} \} \quad (6)$$

As in (Haque et al., 2009), we considered a context window of ±1 and ±2 (i.e. *l*=1, 2) for our experiments.

One natural way of expressing a context-informed feature is as the conditional probability of the target phrase given the source phrase and its context information, as in (7):

$$\hat{h}_m(\hat{f}_k, CI(\hat{f}_k), \hat{e}_k, s_k) = \log P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k)) \quad (7)$$

3.1 Memory-Based Classification

As (Stroppa et al., 2007) point out, directly estimating $P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k))$ using relative frequencies is problematic. Indeed, Zens and Ney (2004) showed that the estimation of $P(\hat{e}_k \mid \hat{f}_k)$ using relative frequencies results in the overestimation of the probabilities of long phrases, so smoothing factors in the form of lexical-based features are often used to counteract this bias (Foster et al., 2006). In the case of context-informed features, since the context is also taken into account, this estimation problem can only become worse. To avoid such problems, in this work we use three memory-based classifiers: IGTree, IB1 and TRIBL [1] (Daelemans et al., 2005). When predicting a target phrase given a source phrase and its context, the source phrase is intuitively the feature with the highest prediction power; in all our experiments, it is the feature with the highest gain ratio (GR).

In order to build the set of examples required to train the classifier, we modify the standard phrase-extraction method of (Koehn et al., 2003) to extract the context of the source phrases at the same time as the phrases themselves. Importantly, therefore, the context extraction comes at no extra cost.

We refer interested readers to (Stroppa et al., 2007) and (Haque et al., 2009) as well as the references therein for more details of how Memory-Based Learning (MBL) is used for classification of source examples for use in the log-linear MT framework.

3.2 Implementation Issues

We split named entities (NE) into characters. We break NEs into transliteration units (TU), which bear close resemblance to syllables. We split English NEs into TUs having C*V* pattern and Hindi NEs are divided into TUs having Ch^+M pattern (M: Hindi *Matra* / vowel modifier, Ch: Characters other than *Matras*). We carry out experiments on both character-level (C-L) and TU-level (TU-L) data. We use a 5-gram language model for all our experiments. The Moses PB-SMT system serves as our baseline system.

The distribution of target phrases given a source phrase and its contextual information is normalised to estimate $P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k))$. Therefore our expected feature is derived as in (8):

$$\hat{h}_{mbl} = \log P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k)) \quad (8)$$

As for the standard phrase-based approach, their weights are optimized using Minimum Error Rate Training (MERT) of (Och, 2003) for each of the experiments.

As (Stroppa et al., 2007) point out, PB-SMT decoders such as Pharaoh (Koehn, 2004) or Moses (Koehn, 2007) rely on a static phrase-table represented as a list of aligned phrases accompanied with several features. Since these fea-

[1] An implementation of IGTree, IB1 and TRIBL is available in the TiMBL software package (http://ilk.uvt.nl/timbl).

tures do not express the context in which those phrases occur, no context information is kept in the phrase-table, and there is no way to recover this information from the phrase-table.

In order to take into account the context-informed features for use with such decoders, the devset and testset that need to be translated are pre-processed. Each token appearing in the test-set and devset is assigned a unique id. First we prepare the phrase table using the training data. Then we generate all possible phrases from the devset and testset. These devset and testset phrases are then searched for in the phrase table, and if found, then the phrase along with its contextual information is given to MBL for classification. MBL produces class distributions according to the maximum-match of the features contained in the source phrase. We derive new scores from this class distribution and merge them with the initial information contained in the phrase table to take into account our feature functions ($\hat{h}_{mbl}$) in the log-linear model (2).

In this way we create a dynamic phrase table containing both the standard and the context-informed features. The new phrase table contains the source phrase (represented by the sequence of ids of the words composing the phrase), target phrase and the new score.

Similarly, replacing all the words by their ids in the development set, we perform MERT using our new phrase table to optimize the feature weights. We translate the test set (words represented by ids) using our new phrase table.

4 Results and Analysis

We used 10,000 NEs from the NEWS 2009 English—Hindi training data (Kumaran and Kellner, 2007) for the standard submission, and the additional English—Hindi parallel person names data (105,905 distinct name pairs) of the Election Commission of India[2] for the non-standard submissions. In addition to the baseline Moses system, we carried out three different set of experiments on IGTree, IB1 and TRIBL. Each of these experiments was carried out on both the standard data and the combined larger data, both at character level and the TU level, and considering ±1/±2 tokens as context. For each experiment, we produce the 10-best distinct hypotheses. The results are shown in Table 1.

We observed that many of the (unseen) TUs in the testset remain untranslated in TU-L systems due to the problems of data sparseness. Whenever a TU-L system fails to translate a TU, we fallback on the corresponding C-L system to translate the TU as a post-processing step.

The accuracy of the TU-L baseline system (0.391) is much higher compared to the C-L baseline system (0.290) on standard dataset. Furthermore, contextual modelling of the source language gives an accuracy of 0.416 and 0.399 for TU-L system and C-L system respectively. Similar trends are observed in case of larger dataset. However, the highest accuracy (0.445) has been achieved with the TU-L system using the larger dataset.

5 Conclusion

In this work, we employed source context modeling into the state-of-the-art log-linear PB-SMT for the English—Hindi transliteration task. We have shown that taking source context into account substantially improve the system performance (an improvement of 43.44% and 26.42% respectively for standard and larger datasets). IGTree performs best for TU-L systems while TRIBL seems to perform better for C-L systems on both standard and non-standard datasets.

Acknowledgements

We would like to thank Antal van den Bosch for his input on the use of memory based classifiers. We are grateful to SFI (http://www.sfi.ie) for generously sponsoring this research under grant 07/CE/I1142.

References

Adimugan Kumaran and Tobias Kellner. A generic framework for machine transliteration. *Proc. of the 30th SIGIR*, 2007.

Byung-Ju Kang and Key-Sun Choi. Automatic transliteration and back-transliteration by decision tree learning. 2000. *Proc. of LREC-2000*, Athens, Greece, pp. 1135-1141.

David Matthews. 2007. Machine Transliteration of Proper Names. Master's Thesis, University of Edinburgh, Edinburgh, United Kingdom.

Franz Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. *Proc. of ACL 2002*, Philadelphia, PA, pp. 295–302.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. *Proc. of EMNLP-2006*, Sydney, Australia, pp. 53-61.

[2] http://www.eci.gov.in/DevForum/Fullname.asp

	S/B	C/TU	Context	ACC	M-F-Sc	MRR	MAP_ref	MAP_10	MAP_sys
Baseline Moses	S	C	0	.290	.814	.393	.286	.131	.131
		TU	0	.391	.850	.483	.384	.160	.160
	B	C	0	.352	.830	.463	.346	.156	.156
		TU	0	.407	.853	.500	.402	.165	.165
IB1	S	C	±1	.391	.858	.501	.384	.166	.166
			±2	.386	.860	.479	.379	.155	.155
		TU	±1	**.406**	.858	.466	.398	.178	.178
			±2	.359	.838	.402	.349	.165	.165
	B	C	±1	**.431**	.865	.534	.423	.177	.177
			±2 (NSD1)	*.420*	*.867*	*.519*	*.413*	*.170*	*.170*
		TU	±1	**.437**	.863	.507	.429	.191	.191
			±2	**.427**	.862	.487	.418	.194	.194
IGTree	S	C	±1	.372	.849	.482	.366	.160	.160
			±2	.371	.847	.476	.364	.156	.156
		TU	±1	**.412**	.859	.486	.404	.164	.164
			±2	**.416**	.860	.493	.409	.166	.166
	B	C	±1	.413	.855	.518	.406	.173	.173
			±2 (NSD2)	*.407*	*.856*	*.507*	*.399*	*.168*	*.168*
		TU	±1	**.445**	.864	.527	.440	.176	.176
			±2	**.427**	.861	.516	.422	.173	.173
TRIBL	S	C	±1	.382	.854	.493	.375	.164	.164
			±2 (SD)	*.399*	*.863*	*.488*	*.392*	*.157*	*.157*
		TU	±1	**.408**	.858	.474	.400	.181	.181
			±2	.395	.857	.453	.385	.182	.182
	B	C	±1	**.439**	.866	.543	.430	.179	.179
			±2 (NSD3)	*.421*	*.864*	*.519*	*.415*	*.171*	*.171*
		TU	±1	**.444**	.863	.512	.436	.193	.193
			±2	**.439**	.865	.497	.430	.197	.197
	S*	C	±2 (NSD4)	*.419*	*.868*	*.464*	*.419*	*.338*	*.338*

Table1: Experimental Results (S/B → Standard / Big data, S*→ TM on Standard data, but LM on Big data, C/TU → Character / TU level, SD→ Standard submission, NSD→ Non-standard submission). Better results with bold faces have not been submitted in the NEWS 2009 Machine Transliteration Shared Task.

Haizhou Li, Zhang Min and Su Jian. 2004. A joint source-channel model for machine transliteration. *Proc. of ACL 2004,* Barcelona, Spain, pp.159-166.

Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. *Computational Linguistics,* 24(4):559-612.

Nicolas Stroppa, Antal van den Bosch and Andy Way. 2007. Exploiting Source Similarity for SMT using Context-Informed Features. *Proc. of TMI-2007*, Skövde, Sweden, pp. 231-240.

Peter F. Brown, S. A. D. Pietra, V. J. D. Pietra and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics* 19 (2), pp. 263-311.

Philipp Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. *Proc. of HLT-NAACL 2003,* Edmonton, Canada, pp. 48-54.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Machine translation: from real users to research: Proc. of AMTA 2004,* Berlin: Springer Verlag, 2004, pp. 115-124.

Philipp Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. *Proc. of ACL,* Prague, Czech Republic, pp. 177-180.

Rejwanul Haque, Sudip Kumar Naskar, Yanjun Ma and Andy Way. 2009. Using Supertags as Source Language Context in SMT. *Proc. of EAMT-09,* Barcelona, Spain, pp. 234-241.

Richard Zens and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. Proc. of *HLT/NAACL 2004*, Boston, MA, pp. 257–264.

Walter Daelemans & Antal van den Bosch. 2005. *Memory-based language processing.* Cambridge, UK, Cambridge University Press.

Yves Lepage and Etienne Denoual. 2006. Objective evaluation of the analogy-based machine translation system ALEPH. *Proc. of the 12th Annual Meeting of the Association of NLP*, pp. 873-876.