Data representation methods and use of mined corpora for Indian language transliteration

Anoop Kunchukuttan **Pushpak Bhattacharyya**
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
{anoopk,pb}@cse.iitb.ac.in

Abstract

Our NEWS 2015 shared task submission is a PBSMT based transliteration system with the following corpus preprocessing enhancements: (i) addition of word-boundary markers, and (ii) language-independent, overlapping character segmentation. We show that the addition of word-boundary markers improves transliteration accuracy substantially, whereas our overlapping segmentation shows promise in our preliminary analysis. We also compare transliteration systems trained using manually created corpora with the ones mined from parallel translation corpus for English to Indian language pairs. We identify the major errors in English to Indian language transliterations by analyzing heat maps of confusion matrices.

1 Introduction

Machine Transliteration can be viewed as a problem of transforming a sequence of characters in one alphabet to another. Transliteration can be seen as a special case of the general translation problem between two languages. The primary differences from the general translation problem are: (i) limited vocabulary size, and (ii) simpler grammar with no reordering. Phrase based statistical machine translation (PB-SMT) is a robust and well-understood technology and can be easily adopted for application to the transliteration problem (Noeman, 2009; Finch and Sumita, 2010). Our submission to the NEWS 2015 shared task is a PBSMT system. Over a baseline PBSMT system, we address two issues: (i) suitable data representation for training, and (ii) parallel transliteration corpus availability.

In many writing systems, the same logical/phonetic symbols can have different character representations depending on whether it occurs in initial, medial or terminal word position. For instance, Indian scripts have different characters for independent vowels and vowel diacritics. Independent vowels typically occurs at the beginning of the word, while diacritics occur in medial and terminal positions. The pronounciation, and hence the transliteration could also depend on the position of the characters. For instance, the terminal *ion* in *nation* would be pronounced differently from initial one in *ionize*. PBSMT learning of character sequence mappings is agnostic of the position of the character in the word. Hence, we explore to transform the data representation to encode position information. Zhang et al. (2012) did not report any benefit from such a representation for Chinese-English transliteration. We investigated if such encoding useful for alphabetic and consonantal scripts as opposed to logographic scripts like Chinese.

It is generally believed that syllabification of the text helps improve transliteration systems. However, syllabification systems are not available for all languages. Tiedemann (2012) proposed a character-level, overlapping bigram representation in the context of machine translation using transliteration. We can view this as weak, coarse and language independent syllabification approach. We explore this overlapping, segmentation approach for the transliteration task.

For many language pairs, parallel transliteration corpora are not publicly available. However, parallel translation corpora like Europarl (Koehn, 2005) and ILCI (Jha, 2012) are available for many language pairs. Transliteration corpora mined from such parallel corpora has been shown to be useful for machine translation, cross lingual information retrieval, etc. (Kunchukuttan et al., 2014). In this paper, we make an intrinsic evaluation of the performance of the automatically mined *BrahmiNet* transliteration corpus (Kunchukuttan et al.,

Proceedings of the Fifth Named Entity Workshop, joint with 53rd ACL and the 7th IJCNLP, pages 78–82,
Beijing, China, July 26-31, 2015.

2015) for transliteration between English and Indian languages. The *BrahmiNet* corpus contains transliteration corpora for 110 Indian language pairs mined from the ILCI corpus, a parallel translation corpora of 11 Indian languages (Jha, 2012).

The rest of the paper is organized as follows. Section 2 and Section 3 describes our system and experimental setup respectively. Section 4 discusses the results of various data representation methods and the use of mined corpus respectively. Section 5 concludes the report.

2 System Description

We use a standard PB-SMT model for transliteration between the various language pairs. It is a discriminative, log-linear model which uses standard SMT features *viz.* direct/inverse phrase translation probabilities, direct/inverse lexical translation probabilities, phrase penalty, word penalty and language model score. The feature weights are tuned to optimize BLEU (Papineni et al., 2002) using the Minimum Error Rate Training algorithm (Och, 2003). It would be better to explore optimizing metrics like accuracy or edit distance instead of using BLEU as a proxy for these metrics. We experiment with various transliteration units as discussed in Section 2.1. We use a 5-gram language model over the transliteration units estimated using Witten-Bell smoothing. Since transliteration does not require any reordering, monotone decoding was done.

2.1 Data Representation

We create different transliteration models based on different basic transliteration units in the source and target training corpus. We use character (P) as well as bigram representations (T). In character based system, the character is the basic unit of transliteration. In bigram-based system, the overlapping bigram is the basic unit of transliteration. We also augmented the word representation with word boundary markers (M) (ˆ for start of word and $ end of word). The various representations we experimented with are illustrated below:

character (P)	H I N D I
character+boundary marker (M)	ˆ H I N D I $
bigram (T)	HI IN ND DI I
bigram+boundary marker (M+T)	ˆH HI IN ND DI I$ $

The abbreviations mentioned above are used subsequently to refer to these data representations.

2.2 Use of mined transliteration corpus

We explore the use of transliteration corpora mined from translation corpora for transliteration. Sajjad et al. (2012) proposed an unsupervised method for mining transliteration pairs from parallel corpus. Their approach models parallel translation corpus generation as a generative process comprising an interpolation of a transliteration and a non-transliteration process. The parameters of the generative process are learnt using the EM procedure, followed by extraction of transliteration pairs from the parallel corpora by setting an appropriate threshold. We compare the quality of the transliteration systems built from such mined corpora with systems trained on manually created NEWS 2015 corpora for English-Indian language pairs.

3 Experimental Setup

For building the transliteration model with the NEWS 2015 shared task corpus as well as the *BrahmiNet* corpus, we used 500 word pairs for tuning and the rest for SMT training. The experimental results are reported on the NEWS 2015 development sets in both cases. The details of the NEWS 2015 shared task datasets are mentioned in shared text report, while the size of the *BrahmiNet* datasets are listed below:

Src	Tgt	Size
En	Hi	10513
En	Ba	7567
En	Ta	3549

We use the *Moses* toolkit (Koehn et al., 2007) to train the transliteration system and the language models were estimated using the SRILM toolkit (Stolcke and others, 2002). The transliteration pairs are mined using the transliteration module in *Moses* (Durrani et al., 2014).

4 Results and Error Analysis

4.1 Effect of Data Representation methods

Table 1 shows transliteration results for various data representation methods on the development set. We see improvements in transliteration accuracy of upto 18% due to the use of word-boundary markers. The MRR also shows an improvement of upto 15%. An analysis of improvement for the En-Hi pair shows that a major reason for the improve-

Src	Tgt	Top-1 Accuracy				F-score				MRR			
		P	M	T	M+T	P	M	T	M+T	P	M	T	M+T
En	Ka	27.6	32.7	28.9	30.4	83.44	85.38	84.75	85.61	39.03	45.15	41.3	41.92
En	Ta	28.6	32.4	31.4	33.4	85.44	86.73	86.64	87.38	41.06	44.89	42.76	45.11
En	Hi	38.82	41.02	37.01	40.52	86.02	86.62	85.77	86.72	51.19	53.28	47.68	51.1
En	He	54.6	56.4	54.4	54.5	91.68	92.29	91.7	91.49	67.68	68.06	64.5	63.76
En	Ba	35.4	38.24	34.48	36.41	86.15	87.13	86	86.78	48.84	51.58	46.56	48.46
Th	En	31.44	32.2	29.64	30.34	84.79	85.09	84.01	84.17	42.6	43.98	40.63	40.48
En	Pe	53.5	57.8	53.3	56.65	91.93	92.76	92.02	92.78	66.58	70.42	64.91	67.66
Ch	En	11.66	10.74	5.33	4.82	72.94	72.33	60.35	61.15	17.95	16.94	8.54	7.52

Table 1: Results on NEWS 2015 development set (in %)

Src	Tgt	P	T
En	Ka	17	25.1
En	Ta	15.3	27.1
En	Hi	27.28	32.3
En	Ba	27.79	32.05
En	He	47.9	54.6
En	Pe	39.35	48.8

Table 2: Top-1 accuracy on NEWS 2015 development set without tuning (in %)

ment seems to the correct generation of vowel diacritics (*maatraa*). Word boundary markers also reduce the following errors: (i) missing initial vowels, (ii) wrong consonants in the initial and final syllable, and (iii) incorrect or spurious generation of *halanta* (inherent vowel suppressor) character. Some examples of these corrections are shown below:

Src	P	M
KALYARI	कालयारी (kAlayArI)	कल्यारी (kalyArI)
NAHAR	नेहर (nehara)	नाहर (nAhara)
AHILYAA	हिल्या (hilyA)	अहिल्या (ahilyA)
AVEDIS	वेडिस (veDisa)	एवेडिस (eveDisa)
AVEDIS	कीर्तपुर (kIrtapura)	कीरतपुर (kIratapura)

We also tried to identify the major errors in English to Indian languages using heat maps of the character-level confusion matrices (Figure 1 shows one for En-Hi). We observed that the following errors are common across all English-Indian language pairs in the shared task: (i) incorrect generation of vowel diacritics, especially confusion between long and short vowels, (ii) *schwa* deletion, (iii) confusion between dental and retroflex consonants, (iv) incorrect or spurious generation of *halanta* (inherent vowel suppressor) character as well as the *aakar maatra* (vowel diacritic for आ(aa)). Hi and Ba show confusion between sibilants (स,श,ष), while Ta and Ka exhibits incorrect or spurious generation of य (ya).

However, the use of a overlapping bigram representation does not show any significant improvement in results over the baseline output. The above results are for systems tuned to maximize BLEU. However, BLEU does not seem the most intuitive tuning metric for the the bigram representation. Hence, we compare the untuned output results (shown in Table 2 for a few language pairs). As we anticipated, we found that the bigram representation gave a significant improvement in accuracy (on an average of about 25%). The combination of word-boundary marker and bigram representation performs best. This suggests the need to tune the SMT system to an alternative metric like edit distance so that the benefit of bigram representation can be properly harnessed. The following is an example where bigram representation resulted in the correct generation of consonants, where the character representation made errors:

Src	P	T
DABHADE	दाबहादे (dAbahAde)	दाभाडे (dAbhADe)

4.2 Transliteration using an automatically mined corpus

Table 3 shows results on the development set when trained using the *BrahmiNet* corpus. The top-1 accuracy is less as compared to training on the NEWS 2015 training corpus. The accuracy very low compared to NEWS 2015 training for Tamil,

Src	Tgt	Accuracy	F-score	MRR
En	Hi	28.39	82.66	39.73
En	Ba	20.59	79.45	30.69
En	Ta	9.3	74.75	15.25

Table 3: Results with BrahmiNet training on NEWS 2105 dev set (in %)

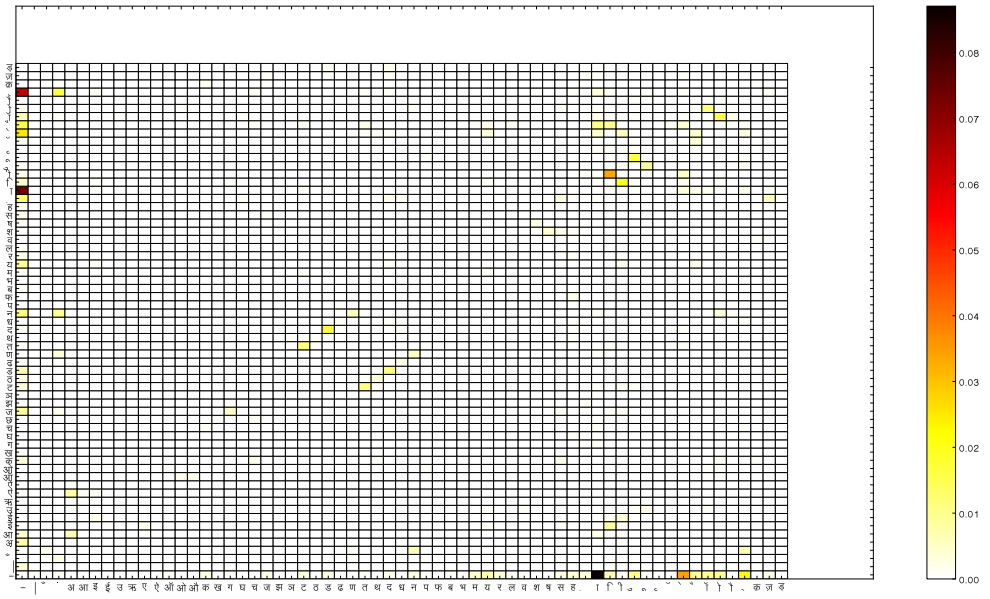

Figure 1: Heat Map for En-Hi (marker, news_2015) system. Color in cell indicates proportion of errors (y-axis: reference set, x-axis: hypothesis set)

where the quality of mined corpus suffers on account of the presence of suffixes due to the agglutinative nature of the language. This results in some wrongly mined pairs as well as smaller number of word pairs being mined. The F-score does not suffer as much as top-1 accuracy and all languages have an F-score greater than 70%. The MRR suggests that the correct transliteration can be found in the top 3 candidates for Hi and Ba, and in the top-7 candidates for Ta. This shows that though the top-1 accuracy of the system is lower than a manually generated corpus, the use of the top-k candidates can be useful in downstream applications like machine translation and cross lingual IR. Since the NEWS 2015 corpus is larger than the *BrahmiNet* corpus, we train a random subset of the NEWS 2015 corpus of the same size as the *BrahmiNet* corpus. In addition, we also experiment with stricter selection thresholds in the mining process.

Since, NEWS 2015 development corpus is quite similar to the NEWS training corpus, we use another corpus (Gupta et al., 2012) to evaluate both the systems. In all these cases, the NEWS corpus gave superior accuracy as compared to *BrahmiNet*. To explain the superiority of the NEWS corpus over all the configurations, we computed the average entropy for the conditional transliteration probability (Chinnakotla et al., 2010). The average entropy for the P(En|Hi) distribution at the character level is higher for the *BrahmiNet* corpus (0.8) as compared to the NEWS 2015 corpus (0.574). The same observation is seen for the P(Hi|En) distribution. This means that there is a higher ambiguity in selecting transliteration in the *BrahmiNet* corpus.

5 Conclusion

We addressed data representation and availability issues in PBSMT based transliteration, with a special focus on English-Indian language pairs. We showed that adding boundary markers to the word representation helps to significantly improve the transliteration accuracy. We also noted that the an overlapping character segmentation can be useful subject to optimizing the appropriate evaluation metrics for transliteration systems. We show that though automatically mined corpora provided lower top-1 transliteration accuracy, the top-10 accuracy, MRR and F-score are competitive to justify the use of the top-k candidates from these mined corpora for translation and IR systems.

References

Manoj Chinnakotla, Om Damani, and Avijit Satoskar. 2010. Transliteration for resource-scarce languages. *ACM Transactions on Asian Language Information Processing (TALIP)*.

Nadir Durrani, Hieu Hoang, Philipp Koehn, and Hassan Sajjad. 2014. Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. *EACL 2014*.

Andrew M Finch and Eiichiro Sumita. 2010. A bayesian model of bilingual segmentation for transliteration. In *IWSLT*.

Kanika Gupta, Monojit Choudhury, and Kalika Bali. 2012. Mining hindi-english transliteration pairs from online hindi lyrics. In *LREC*, pages 2459--2465.

Girish Nath Jha. 2012. The TDIL program and the Indian Language Corpora Initiative. In *Language Resources and Evaluation Conference*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177--180. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79--86.

Anoop Kunchukuttan, Ratish Pudupully, Rajen Chatterjee, Abhijit Mishra, and Pushpak Bhattacharyya. 2014. The IIT Bombay SMT System for ICON 2014 Tools Contest. In *NLP Tools Contest at ICON 2014*.

Anoop Kunchukuttan, Ratish Puduppully, and Pushpak Bhattacharyya. 2015. Brahmi-Net: A transliteration and script conversion system for languages of the Indian subcontinent. In *Conference of the North American Chapter of Association for Computational Linguistics: System Demonstrations*.

Sara Noeman. 2009. Language independent transliteration system using phrase based smt approach on substrings. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311--318. Association for Computational Linguistics.

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

Jörg Tiedemann. 2012. Character-based pivot translation for under-resourced languages and domains. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 141--151. Association for Computational Linguistics.

Chunyue Zhang, Tingting Li, and Tiejun Zhao. 2012. Syllable-based machine transliteration with extra phrase features. In *Proceedings of the 4th Named Entity Workshop*.