Leveraging Transliterations from Multiple Languages

Aditya Bhargava, Bradley Hauer, and **Grzegorz Kondrak**
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
{ab31,bmhauer,gkondrak}@ualberta.ca

Abstract

While past research on machine transliteration has focused on a single transliteration task, there exist a variety of supplemental transliterations available in other languages. Given an input for English-to-Hindi transliteration, for example, transliterations from other languages such as Japanese or Hebrew may be helpful in the transliteration process. In this paper, we propose the application of such supplemental transliterations to English-to-Hindi machine transliteration via an SVM re-ranking method with features based on n-gram alignments as well as system and alignment scores. This method achieves a relative improvement of over 10% over the base system used on its own. We further apply this method to system combination, demonstrating just under 5% relative improvement.

1 Introduction

The focus of significant previous work in machine transliteration, including that presented at past NEWS Shared Tasks (Li et al., 2009; Kumaran et al., 2010b), has been on single transliteration tasks in isolation of other other languages. This is despite the fact that the various languages provided represent a significant quantity of potentially useful data that is being ignored. In this NEWS 2011 Shared Task submission, we present a method which beneficially applies supplemental transliterations from other languages to English-to-Hindi transliteration.

In practice, this is a realistic situation in which transliterations from other languages can help. For example, Wikipedia contains articles on guitarist John Petrucci in English and Japanese, but not in Hindi. If we wanted to automatically generate a stub (skeleton) article in Hindi, we would need to transliterate his name into Hindi. Since a Japanese version already exists, we could extract from it additional information to help with the transliteration process. Importantly, since our article is about an American guitarist, we would explicitly want to start with the English (original) version of the name, and treat other languages as extra data, rather than vice versa.

In order to effectively incorporate the other-language data, we apply SVM re-ranking in a manner that has previously been shown to provide significant improvement for grapheme-to-phoneme conversion (Bhargava and Kondrak, 2011). This method is flexible enough to incorporate multiple languages; it employs features based on character alignments between potential outputs and existing transliterations from other languages, as well as scores of these alignments, which serve as a measure of similarity. We apply this approach on top of the same DIRECTL+ system as submitted last year (Jiampojamarn et al., 2010b) for English-to-Hindi machine transliteration. Compared to the base DIRECTL+ performance, we are able to achieve significantly better results, with a relative performance increase of over 10%. We also achieve improvements without supplemental transliterations by simply apply the same approach with another *system*'s output as extra data. We furthermore experiment with romanization for Hindi data as well as different alignment length settings for English-to-Chinese transliteration. This paper presents methods, methodology, and results for the above experiments.

2 Leveraging multiple transliterations

Bhargava and Kondrak (2011) present a method for applying transliterations to grapheme-to-phoneme conversion. Here, we apply this method verbatim to machine transliteration. The method is based on SVM re-ranking applied over n-best output lists generated by a base system. Intuitively, we have

Proceedings of the 2011 Named Entities Workshop, IJCNLP 2011, pages 36–40,
Chiang Mai, Thailand, November 12, 2011.

an existing base transliteration system that, for a given input, provides a set of n scored outputs, with the correct output not always appearing in the top position. In order to help bring the correct output to the top, we turn to existing transliterations of the input *from other languages*. In order to leverage a variety of features and transliterations from all available languages, SVM re-ranking is applied to this task.

For each output, a feature vector is constructed. Given alignments between the input and output, for example, binary indicator features based on grouping input and output n-grams in the style of DIRECTL+ (Jiampojamarn et al., 2010a) are constructed. The base system's score for the output would be included as well, along with differences between the given output's score and the scores for the other outputs in the list. This feature construction process is then repeated, replacing the input with an available transliteration, for each available transliteration language. The score in this latter case is used as a measure of how "similar" a candidate output is to a "reference" transliteration from another language. We refer to these other transliterations as *supplemental* transliterations. While the score features provide a global measure of similarity, the n-gram features allow weights to be learned for character combinations between the candidate output and supplemental transliterations; this provides very fine-grained features that can explicitly use certain characters in supplemental transliterations to help determine the quality of a candidate output.

There are, however, some practicalities that must be considered. Bhargava and Kondrak (2011) note the importance of applying multiple languages; they found it difficult to achieve significant improvements using transliterations from one language only. This is due in part to noise in the data (which has been observed in some of the NEWS Shared Task data (Jiampojamarn et al., 2009)) as well as differing conventions for various transliteration "schemes". These issues are handled implicitly in two ways: (1) the granularity of the n-gram features allows certain character combinations in the transliteration to be learned as being positive or negative indicators of a candidate output's quality, or that they should be ignored altogether; and (2) the use of multiple transliterations helps smooth out some of the noise. While we do not examine these methods here for brevity's sake, Bhargava and Kondrak (2011) show the effectiveness of the granular n-gram features vs. the score features as well as the importance of applying multiple transliteration languages.

3 Alignment of training data

Practically, we must consider how to generate the alignments between the candidate output transliterations and the supplemental transliterations for the n-gram features, as well as how to generate the similarity scores. M2M-ALIGNER (Jiampojamarn et al., 2007) addresses both of these. M2M-ALIGNER is an unsupervised character alignment system, meaning that it can learn to align data given sufficient training data consisting of unaligned input-output pairs. Once trained, M2M-ALIGNER will then produce an alignment for a new pair as well as an alignment score. Because the algorithm is a many-to-many extension of the unsupervised edit distance algorithm, we can see that the alignment score should represent some notion of script-agnostic similarity.

Since we will be applying M2M-ALIGNER between candidate output transliterations and supplemental transliterations for a variety of supplemental languages, we will need to build several alignment models, each being built from separate training data. The majority of the task data are English-source, so for any entry in one language corpus we can easily find corresponding transliterations in other language corpora. In other words, to generate training data for M2M-ALIGNER between the target transliteration language and a supplemental language, we need only intersect the two corpora on the basis of the common English input.

Table 1 shows the amount of overlap between the test data for the different English-source languages and the combined training and development data for the other English-source languages. Note that the Chinese- and Korean-target corpora show very high coverage; however, we focus on English-to-Hindi transliteration as it enables us to more closely examine the outputs based on our own linguistic familiarities. The use of other corpora here requires that these results be submitted as a non-standard run. Note that, because there is not complete coverage for the English-to-Hindi test data, we simply submit the base system's results as-is in cases where there is no transliteration available from other languages.

Language	Test set	Overlap
EnBa	1,000	498
EnCh	2,000	2,000
EnHe	1,000	525
EnHi	1,000	889
EnJa	1,815	734
EnKa	1,000	883
EnKo	609	608
EnPe	2,000	1,049
EnTa	1,000	884
EnTh	2,000	1,564

Table 1: The number of entries in the test data (per language) that have at least one supplemental transliteration available from another language corpus.

4 Base systems

Our principal base system that generates the n-best output lists is DIRECTL+, which has produced excellent results in the NEWS 2010 Shared Task on Transliteration (Jiampojamarn et al., 2010b). For re-ranking, note that training a re-ranker requires training data where the base system scores are representative of unseen data so that the re-ranker does not simply learn to follow the base system; we therefore split the training data into ten folds and perform a sort-of cross validation with DIRECTL+. This provides us with usable training data for re-ranking. We tune the SVM's hyperparameter based on performance on the provided development data, and use the best DIRECTL+ settings established in the NEWS 2010 Shared Task (Jiampojamarn et al., 2010b). Armed with optimal parameter settings, we combine the training and development data into a single set used to train our final DIRECTL+ system. We also repeat the cross-validation process for training the re-ranker.

We also apply the SVM re-ranking approach to system combination. In this case, we additionally train another system—here we use SEQUITUR (Bisani and Ney, 2008)—for English-to-Hindi transliteration. During test time, we feed the input into *both* DIRECTL+ and SEQUITUR, and use the top SEQUITUR output as supplemental data. We expect that sometimes SEQUITUR will provide a correct answer where DIRECTL+ does not; the hope is that the SVM re-ranking approach will be able to learn when this is the case based on the n-gram and score features.

Language	Type	System	Acc.
EnHi	Standard	DTL	47.1
EnHi	Standard	DTL+Rom.	45.7
EnHi	Standard	DTL+SEQ	49.3
EnHi	Non-Std.	DTL+Supp.	52.1
EnCh	Standard	DTL 3-1	34.1
	Standard	DTL 7-1	28.7
EnJa	Standard	DTL	43.5

Table 2: Word accuracy (%) for the various submitted runs. DTL is generic DIRECTL+; DTL+Rom. is DIRECTL+ trained on romanized data; DTL+SEQ is DIRECTL+ re-ranked with SEQUITUR outputs; and DTL+Supp. is DIRECTL+ re-ranked with supplemental transliteration data from other languages.

5 Hindi romanization

In addition to the above re-ranking approach, we experimented with a romanization method for the Hindi data. Since consonant characters in the Devanagari alphabet have vowels included by default, we romanize the text in order to provide DIRECTL+ with direct individual control over the consonant and vowel components of the Hindi characters. The default vowel is changed by means of diacritic-like characters, which in turn deletes the default vowel; this requires a context-sensitive (but still rule-based) romanization method, which we construct manually. We then train DIRECTL+ on the romanized data; during testing, we take the romanized output and convert it back into Devanagari Unicode characters, again using a manually-constructed context-sensitive rule-based converter.

6 Results

Table 2 shows that SVM re-ranking significantly improves the English-to-Hindi transliteration accuracy in comparison with the base system. Leveraging all of the English-source transliteration corpora as supplemental data yields an increase of over 10%. When applied using SEQUITUR's output as "supplemental" data, we see almost a 5% (relative) increase in word accuracy.

In contrast, our Hindi romanization approach decreases the accuracy. This differs from the results of the successful application of romanization to Japanese (Jiampojamarn et al., 2010b), demonstrating that it is not always possible to transfer an idea

from one language to another.

The English-to-Chinese results, which use only the base DIRECTL+ system, demonstrate the importance of the alignment length parameter setting. DIRECTL+ requires aligned data for input, and the maximum length of the alignments will have an effect on what DIRECTL+ learns to produce. We submitted both 3-to-1 and 7-to-1 alignments because they gave similar results during development, and both were better than other tested possibilities. In the final results, we see a substantial difference between the two alignment settings. We hypothesize that the complexity of English-to-Chinese mappings is better captured by the alignments that map longer sequences of English letters to single Chinese characters. making it difficult to generalize to new data.

Finally, we observe very good overall accuracy in the English-to-Japanese results (which also only use base DIRECTL+), which further confirm the effectiveness of DIRECTL+ when applied to machine transliteration.

7 Previous work

There are three lines of research that are relevant to the work we have presented in this paper: (1) DIRECTL+ and SEQUITUR for machine transliteration; (2) applying multiple languages; and (3) system combination.

For the NEWS 2009 and 2010 Shared Tasks, the discriminative DIRECTL+ system that incorporates many-to-many alignments, online max-margin training and a phrasal decoder was shown to function well as a general string transduction tool; while originally designed for grapheme-to-phoneme conversion, it produced excellent results for machine transliteration (Jiampojamarn et al., 2009; Jiampojamarn et al., 2010b), leading us to re-use it here. Finch and Sumita (2010) also submitted a top-performing system that was based in part on SEQUITUR, which is a generative system based on joint n-gram modelling (Bisani and Ney, 2008).

In this paper, we applied multiple transliteration languages to a single transliteration task. While our method is based on SVM re-ranking with similar features as to those used in the base system (Bhargava and Kondrak, 2011), there have been other explorations into incorporating other language data, particularly when data are scarce. Zhang et al. (2010), for example, apply a pivoting approach to machine transliteration, and similarly Khapra et al. (2010) propose to transliterate through "bridge" languages. Along similar lines, Kumaran et al. (2010a) find increases in accuracy using a linear-combination-of-scores system that combined the outputs of a direct transliteration system with a system that transliterated through a third language. For statistical machine translation, Cohn and Lapata (2007) also explore the use of a third language.

Finally, we also touched briefly on system combination: we applied the SVM re-ranking method to combining the outputs of both DIRECTL+ and SEQUITUR, in particular treating DIRECTL+ as the base system and using SEQUITUR's best outputs to re-rank DIRECTL+'s output lists. Finch and Sumita (2010), in contrast, combine SEQUITUR's output with that of a phrase-based statistical machine translation system, achieving excellent results. Where our approach is based on SVM re-ranking, theirs merged the outputs of the two systems together and then used a linear combination of the system scores to re-rank the combined list.

8 Conclusion

In this paper, we described our submission to the NEWS 2011 Shared Task on machine transliteration. Our focus was on incorporating supplemental data, using a method based on SVM re-ranking, with features derived from n-gram alignments and alignment scores. We demonstrated improvements of over 10% when applying other transliteration data to English-to-Hindi machine transliteration, and just under 5% when applying another system's outputs in a similar manner. We also found that the romanization of Hindi characters brings about a decrease in performance, and that the alignment length parameter in the DIRECTL+ system has a critical effects on the results.

Acknowledgements

We are grateful to Ying Xu for examining our initial Chinese results. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Aditya Bhargava and Grzegorz Kondrak. 2011. How do you pronounce your name? Improving G2P with transliterations. In *Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, June. Association for Computational Linguistics.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, May.

Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735, Prague, Czech Republic, June. Association for Computational Linguistics.

Andrew Finch and Eiichiro Sumita. 2010. Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model. In *Proceedings of the 2010 Named Entities Workshop*, pages 48–52, Uppsala, Sweden, July. Association for Computational Linguistics.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379, Rochester, New York, USA, April. Association for Computational Linguistics.

Sittichai Jiampojamarn, Aditya Bhargava, Qing Dou, Kenneth Dwyer, and Grzegorz Kondrak. 2009. DirecTL: a language independent approach to transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 28–31, Suntec, Singapore, August. Association for Computational Linguistics.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010a. Integrating joint n-gram features into a discriminative training framework. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 697–700, Los Angeles, California, USA, June. Association for Computational Linguistics.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. 2010b. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47, Uppsala, Sweden, July. Association for Computational Linguistics.

Mitesh M. Khapra, A Kumaran, and Pushpak Bhattacharyya. 2010. Everybody loves a rich cousin: An empirical study of transliteration through bridge languages. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 420–428, Los Angeles, California, June. Association for Computational Linguistics.

A. Kumaran, Mitesh M. Khapra, and Pushpak Bhattacharyya. 2010a. Compositional machine transliteration. *ACM Transactions on Asian Language Information Processing (TALIP)*, 9(4):13:1–29, December.

A. Kumaran, Mitesh M. Khapra, and Haizhou Li. 2010b. Report of NEWS 2010 Transliteration Mining Shared Task. In *Proceedings of the 2010 Named Entities Workshop*, pages 21–28, Uppsala, Sweden, July. Association for Computational Linguistics.

Haizhou Li, A. Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of NEWS 2009 Machine Transliteration Shared Task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 1–18, Suntec, Singapore, August. Association for Computational Linguistics.

Min Zhang, Xiangyu Duan, Vladimir Pervouchine, and Haizhou Li. 2010. Machine transliteration: Leveraging on third languages. In *Coling 2010: Posters*, pages 1444–1452, Beijing, China, August. Coling 2010 Organizing Committee.