Multiple System Combination for Transliteration

Garrett Nicolai, Bradley Hauer, Mohammad Salameh,
Adam St Arnaud, Ying Xu, Lei Yao, Grzegorz Kondrak
Department of Computing Science
University of Alberta, Edmonton, Canada
{nicolai,bmhauer,msalameh,ajstarna,
yx2,lyao1,gkondrak}@ualberta.ca

Abstract

We report the results of our experiments in the context of the NEWS 2015 Shared Task on Transliteration. We focus on methods of combining multiple base systems, and leveraging transliterations from multiple languages. We show error reductions over the best base system of up to 10% when using supplemental transliterations, and up to 20% when using system combination. We also discuss the quality of the shared task datasets.

1 Introduction

The 2015 NEWS Shared Task on Machine Transliteration continues the series of shared tasks that were held yearly between 2009 and 2012. With the exception of the 2010 edition that included transliteration mining, the task has been limited to learning transliteration models from the training sets of word pairs. Participants are allowed to use target lexicons or monolingual corpora, but since those are "non-standard", the results are not comparable across different teams. Another drawback of the current framework is the lack of context that is required to account for morphological alterations.

Our University of Alberta team has participated in each of the five editions of this shared task. Although this year's task is virtually identical to the 2012 task, there has been progress in transliteration research since then. In particular, transliteration projects at the University of Alberta have led to the design of novel techniques for leveraging supplemental information such as phonetic transcriptions and transliterations from other languages. During those projects, we also observed that combinations of diverse systems often outperform their component systems. We decided to test this hypothesis in the current rerun of the NEWS shared task.

In this paper, we describe experiments that involve three well-known transliteration approaches. DIRECTL+, SEQUITUR, and statistical machine translation toolkits (SMT). In an effort to harness the strengths of each system, we explore various techniques of combining their outputs. Furthermore, we experiment with leveraging transliterations from other languages, in order to test whether this can improve the overall results. We obtain state-of-the-art results on most language pairs.

2 Base Systems

In this section, we describe our three base systems: DIRECTL+, SEQUITUR, and SMT.

2.1 DirecTL+

DIRECTL+ is a publicly-available[1] discriminative string transduction tool, which was initially developed for grapheme-to-phoneme conversion (Jiampojamarn et al., 2008). DIRECTL+ was successfully applied to transliteration in the previous NEWS shared tasks by our team (Jiampojamarn et al., 2009; Jiampojamarn et al., 2010b; Bhargava et al., 2011; Kondrak et al., 2012), as well as by other teams (Okuno, 2012; Wu et al., 2012). We make use of all features described by Jiampojamarn et al. (2010a). We perform source-target pair alignment with *mpaligner* (Kubo et al., 2011) because it performed slightly better in our development experiments than M2M-aligner (Jiampojamarn et al., 2007). The parameters of the transducer and the aligner were tuned separately for each language pair.

2.2 SEQUITUR

SEQUITUR is a joint n-gram-based string transduction system[2] originally designed for grapheme-to-phoneme transduction (Bisani and Ney, 2008), which is also applicable to a wide

[1] *https://code.google.com/p/directl-p*
[2] *http://www-i6.informatik.rwth-aachen.de/web/Software*

variety of monotone translation tasks including transliteration (Finch and Sumita, 2010; Nejad et al., 2011). Unlike DIRECTL+, which requires aligned source-target pairs, SEQUITUR directly trains a joint n-gram model for transduction from unaligned data. Higher order n-gram models are trained iteratively: a unigram model is created first; this model is then used to train a bigram model, which is then in turn used to train a trigram model, and so on. The order of the model trained is a parameter tuned on a development set.

An important limitation of SEQUITUR is that both the source and target character sets are limited to a maximum of 255 symbols each. This precludes a direct application of SEQUITUR to scripts such as Chinese, Korean, and Japanese Kanji. Ultimately, it was a factor in our decision to leave out the datasets that involve these languages.

2.3 SMT

We frame the transliteration task as a machine translation task by treating individual characters as words, and sequences of characters as phrases. We align the word pairs with GIZA++ (Och and Ney, 2003), and use Moses (Koehn et al., 2007), a phrase-based SMT system, to generate transliterations. The decoder's log-linear model includes a standard feature set. Four translation model features encode phrase translation probabilities and lexical scores in both directions. Both alignment and generation are monotonic, i.e. reordering is disabled, with distortion limit set to zero. We train a KN-smoothed 5-gram language model on the target side of the parallel training data with SRILM (Stolcke, 2002). If a source word is provided with several target transliterations, we select the first one. The decoder's log-linear model is tuned with MERT (Och, 2003). We use BLEU score (Papineni et al., 2002) as an evaluation metric during tuning.

3 Language-specific Preprocessing

Our development experiments showed that romanization of Chinese and Japanese characters can be helpful.

For the alignment of English and Chinese (EnCh) names, we convert the Chinese names in the training data into Pinyin romanization, as described in Kondrak et al. (2012). This set of training pairs is aligned using our many-to-many aligner, and the resulting alignment links are projected onto Chinese characters. In cases where alignments split individual Chinese characters, they are expanded to include the entire character. Finally, the generation model is derived from the alignment between English letters to Chinese characters.

For English-to-Japanese (EnJa) transliteration, the Katakana symbols are first converted to Latin characters following a deterministic mapping, as described in Jiampojamarn et al. (2009). The English characters are aligned to the Latin characters, and a generation model is learned from the alignments. At test time, the model outputs Latin symbols, which are converted back into Japanese Katakana. We employed a similar approach for SEQUITUR.

4 System Combination

Each of our base systems can generate n-best lists of predictions, together with confidence scores. We experimented with several methods of combining the outputs of the base systems.

4.1 LINCOMB

We generate the n-best transliterations for each test input, and combine the lists via a linear combination of the confidence scores. Scores are first normalized according to the following formula:

$$normScore = \frac{(score - minScore)}{(maxScore - minScore)}$$

where $minScore$ is the confidence score of the n-th best prediction, and $maxScore$ is the confidence score of the best prediction. Predictions that do not occur in a specific system's n-best predictions are also given a score of 0 for combination. n is set to 10 in all of our experiments. If an n-best list contains less than 10 predictions, $minScore$ is set to the score of the last prediction in the list. Our development experiments indicated that this method of combination was more accurate than a simpler method that uses only the prediction ranks.

4.2 RERANK

Bhargava and Kondrak (2012) propose a reranking approach to transliteration to leverage supplemental representations, such as phonetic transcriptions and transliterations from other languages. The reranker utilizes many features, including the similarity of the candidate outputs to the supplemental

representations, several types of n-gram features, and the confidence scores of the base system itself. Once a feature vector is created for each output, weights are learned with an SVM reranker.

Bhargava et al. (2011) apply the reranking approach (RERANK) to system combination. The idea is to rerank the n-best list output from a base system, using the top prediction from another system. If the correct output is in the n-best list, reranking has the potential to elevate it to the top. The paper reports a 5% relative increase in accuracy on EnHi with DIRECTL+ and SEQUITUR as the base and supplemental system, respectively.

For this shared task, we investigated two modifications of RERANK. First, we attempted to extend the original approach to take advantage of more than one supplemental system. For this purpose, we experimented with *cascaded reranking*, in which the n-best list is reranked using the top outputs of both supplemental systems in turn. Second, in an attempt to emulate the effectiveness of the linear combination approach, we experimented with restricting the set of features to confidence scores from the individual systems.

4.3 JOINT

Yao and Kondrak (2015) propose a JOINT generation approach that can incorporate multiple transliterations as input, and show that it outperforms the reranking approach of Bhargava and Kondrak (2012). The JOINT system is a modified version of DIRECTL+ that utilizes aligned supplemental transliterations to learn additional features. Supplemental transliterations are then provided to the system at test time, in order to generate the final output.

For this shared task, we performed two sets of experiments with the JOINT system. While the JOINT system was designed to incorporate additional transliterations as supplemental information, we were also interested if it could be used for system combination. For this purpose, we provided the JOINT system with the output of all three base systems as supplemental inputs. In addition, we experimented with attaching distinct tags to each character in the supplemental inputs, in order to make a distinction between the symbols produced by different supplemental systems. The JOINT system was trained on a held-out set composed of the outputs of the base systems generated for each source word.

	DTL	SEQ	SMT	LINCOMB
ArEn	51.4	45.9	47.1	**57.1**
EnBa	37.1	37.8	34.9	**40.1**
EnCh	29.4	–	27.9	**29.7**
EnHe	**61.3**	56.6	53.1	60.1
EnHi	43.5	40.4	36.8	**45.4**
EnJa	38.9	35.8	31.8	**40.3**
EnKa	32.7	35.7	28.1	**37.4**
EnPe	**34.7**	32.0	29.0	34.6
EnTa	**38.5**	34.4	29.3	38.4
EnTh	36.2	35.8	30.6	**39.5**
ThEn	33.2	36.5	34.3	**39.5**

Table 1: Transliteration accuracy of DIRECTL+, SEQUITUR, and SMT on the development sets.

The second set of experiments followed the original design of Yao and Kondrak (2015), in which the supplemental data consists of transliterations of a source word in other languages. We extracted the supplemental transliterations from the NEWS 2015 Shared Task training and development sets for which English was the source language. For words with no supplemental transliterations, we fall back on base DIRECTL+ output.

5 Development Experiments

For our development experiments, we randomly split the provided training sets into ten equal folds, of which eight were used for base system training, and one for base system tuning, with the final fold held out for system combination training. The base models were trained without language-specific preprocessing.

Table 1 shows the results on the provided development set. DIRECTL+ is the best performing base system on eight datasets, with SEQUITUR winning on the remaining three. Although SMT is never the best, it comes second on three tasks. The absolute differences between the three system are within 10%.

Because of its simplicity, we expected LINCOMB to serve as the baseline combination method. However, as shown in Table 1, it performs surprisingly well, providing an improvement over the best base system on eight out of eleven datasets. An additional advantage of LINCOMB is that it requires no training or parameter tuning. Since the other two combination methods are more complicated and less reliable, we chose LINCOMB as our default method.

	NEWS 2011	NEWS 2012
ArEn	**61.7**	**59.6**
EnBa	**50.9**	**49.2**
EnCh	33.2	31.4
EnHe	**62.2**	**18.0**
EnHi	**48.8**	64.9
EnJa	**42.5**	39.7
EnKa	**43.4**	54.5
EnPe	36.1	**71.0**
EnTa	**47.7**	58.5
EnTh	**41.0**	14.1
ThEn	27.3	**15.6**

Table 2: Official test results for standard linear combination (LINCOMB).

Some configurations of RERANK did achieve improvements over the best base system on most sets, but the results were generally below LINCOMB. This confirms the observation of (Bhargava and Kondrak, 2012) that LINCOMB is a strong combination baseline because it utilizes entire n-best lists from all systems.

The JOINT approach was unable to improve over base DIRECTL+ when trained on relatively small held-out sets. We also tried to leverage the entire training set for this purpose using 10-cross validation. However, that method requires a substantial amount of time and computing resources, and after disappointing initial results on selected datasets, we decided to forgo further experimentation. It remains an open question whether the joint generation approach can be made to work as a system combination.

The JOINT approach performs much better in its original setup, in which additional transliterations from other languages are provided as input. However, its effectiveness depends on the amount of supplemental information that is available per source word. The improvement of JOINT over base DIRECTL+ seems to be correlated with the percentage of words with at least two supplemental transliterations in the corresponding test set. The language pairs with over 50% of such words in the development set include EnHi, EnKa, and EnTa.

6 Test Results

Table 2 shows the official test results for LINCOMB. Following our development results, we designated LINCOMB for our primary runs except on EnHe, EnPe, and EnTa, where DIRECTL+ was chosen instead (see the results in Table 3). Overall, our standard runs achieved top results on 14 out of 22 datasets.

	NEWS 2011		NEWS 2012	
	DTL	JOINT	DTL	JOINT
EnHe	**62.2**	61.6	17.4	**18.4**
EnHi	47.7	**53.1**	55.8	55.9
EnKa	42.5	**44.1**	47.5	49.1
EnTa	47.6	**48.0**	53.7	52.8
EnPe	**38.2**	–	68.3	–

Table 3: Official test results for standard DIRECTL+, and for non-standard JOINT with supplemental transliterations.

Table 3 includes our remaining test results. We submitted the JOINT runs on languages that had promising improvements in the development results. These runs were designated as non-standard even though the supplemental transliterations are from the provided NEWS datasets. For these languages, we also submitted standard DIRECTL+ runs, in order to gauge the improvement obtained by JOINT. The JOINT outperformed base DIRECTL+ on six out of eight datasets.

We observe many cases where the test results diverge from our development results. It appears that the provided development sets are not always representative of the final sets. To give some examples, the 2012 ArEn test set contains only a single space, as compared to 878 spaces present on the source side of the corresponding development set, while one-third of the target-side characters in the EnCh development set do not occur at all in the corresponding training set. In addition, the 2011 and 2012 test sets vary wildly in difficulty, as evidenced by the results in Table 2.

7 Conclusion

We found that simple linear combination of normalized confidence scores is an effective and robust method of system combination, although it is not guaranteed to improve upon the best base system. We also showed that a joint generation approach that directly leverages supplemental transliterations has the potential of boosting transliteration accuracy. However, the generality of these conclusions is limited by the narrow scope of the shared task and the deficiencies of the provided datasets.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada, and the Alberta Innovates – Technology Futures.

References

Aditya Bhargava and Grzegorz Kondrak. 2012. Leveraging supplemental representations for sequential transduction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 396–406, Montréal, Canada.

Aditya Bhargava, Bradley Hauer, and Grzegorz Kondrak. 2011. Leveraging transliterations from multiple languages. In *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, pages 36–40, Chiang Mai, Thailand.

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

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.

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.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proceedings of ACL-08: HLT*, pages 905–913, Columbus, Ohio.

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.

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.

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Grzegorz Kondrak, Xingkai Li, and Mohammad Salameh. 2012. Transliteration experiments on Chinese and Arabic. In *4th Named Entity Workshop (NEWS)*, pages 71–75, Jeju, Korea. System paper.

Keigo Kubo, Hiromichi Kawanami, Hiroshi Saruwatari, and Kiyohiro Shikano. 2011. Unconstrained many-to-many alignment for automatic pronunciation annotation. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Xi'an, China.

Najmeh Mousavi Nejad, Shahram Khadivi, and Kaveh Taghipour. 2011. The Amirkabir machine transliteration system for NEWS 2011: Farsi-to-English task. In *2011 Named Entities Workshop*, page 91.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July.

Yoh Okuno. 2012. Applying mpaligner to machine transliteration with Japanese-specific heuristics. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, pages 61–65, Jeju, Korea.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Intl. Conf. Spoken Language Processing*, pages 901–904.

Chun-Kai Wu, Yu-Chun Wang, and Richard Tzong-Han Tsai. 2012. English-Korean named entity transliteration using substring alignment and reranking methods. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, pages 57–60, Jeju, Korea.

Lei Yao and Grzegorz Kondrak. 2015. Joint generation of transliterations from multiple representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 943–952, Denver, Colorado.