Boosting English-Chinese Machine Transliteration via High Quality Alignment and Multilingual Resources

Yan Shao, Jörg Tiedemann, Joakim Nivre
Department of Linguistics and Philology
Uppsala University
{yan.shao, jorg.tiedemann, joakim.nivre}@lingfil.uu.se

Abstract

This paper presents our machine transliteration systems developed for the NEWS 2015 machine transliteration shared task. Our systems are applied to two tasks: English to Chinese and Chinese to English. For standard runs, in which only official data sets are used, we build phrase-based transliteration models with refined alignments provided by the M2M-aligner. For non-standard runs, we add multilingual resources to the systems designed for the standard runs and build different language specific transliteration systems. Linear regression is adopted to rerank the outputs afterwards, which significantly improves the overall transliteration performance.

1 Introduction

Machine transliteration is an effective approach to process named entities that are out-of-vocabulary words in many NLP tasks, such as machine translation, corpus alignment and cross-language information retrieval. In this paper, using the experiment data from the NEWS 2015 machine transliteration shared task (Zhang et al., 2015), we develop machine transliteration systems respectively targeting English to Chinese and Chinese to English transliteration tasks.

The M2M-aligner (Jiampojamarn et al., 2007) is used to preprocess the training data to obtain the boundaries and alignments of transliteration units between source and target language. We apply a hard-constrained estimation-maximization (EM) algorithm to post-process its outputs, which greatly reduces errors of segmentation and alignment. With the refined outputs, we build phrase-based transliteration systems using Moses (Koehn et al., 2007), a popular statistical machine translation framework. The results are submitted as standard runs.

Since transliteration is the transcription preserving the pronunciation of the source language, source names that are written in the same script can be pronounced differently in different language and therefore the transliterations will not be the same. Thus, we build various language specific transliteration systems using multilingual resources. Linear regression is used to rerank the outputs, where the individual scores of translation models in Moses are used as features. The results are submitted as non-standard runs.

2 Background

Machine transliteration is often modelled as a sequence labelling problem in previous research. Thus, the existing algorithms for sequence labelling all can be used for solving the problem. The classical joint source-channel model (Li et al., 2004) is essentially a Hidden Markov Model (HMM), which allows direct mapping between the transliteration units in source and target languages. Given the source string as the input, when it passes through the joint source-channel, the output is generated simultaneously.

Chen et al. (2011) extends the original source-channel model into multi-to-multi source-channel model and uses Moses as the decoder. As a popular experimental framework for machine translation, Moses is also applied to build phrase-based transliteration systems in some other related works (Finch and Sumita, 2010). Machine transliteration is treated as character level machine translation without distortion in their approaches.

In addition, the use of Conditional Random Fields (CRF) (Lafferty et al., 2001) is another popular approach in previous studies. It is a powerful discriminative sequence labelling model that uses rich local features. However, it is very costly in terms of time complexity during the training process especially combined with the full transliteration task. Qin and Chen (2011) decomposes the

Proceedings of the Fifth Named Entity Workshop, joint with 53rd ACL and the 7th IJCNLP, pages 56–60,
Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

full task into several subtasks and uses different CRF recognizers. Kuo et al. (2012) uses a two-stage CRF system with accessor variety (AV) as an additional feature, which processes segmentation and mapping separately.

3 System Description

3.1 Preprocessing Training Data

As in the case of machine translation, the training data for constructing transliteration systems usually do not contain required alignments between source and target languages. In this study, we use the M2M-Aligner to preprocess the training data and obtain the boundaries and alignment information of transliteration units. The M2M-Aligner uses an EM algorithm, which is an extension of the forward-backward training of the one-to-one stochastic transducer originally presented by Ristad and Yianilos (1998).

Since the performance of the aligner has a great impact on the overall transliteration quality, we preprocess the M2M-Aligner's input as well as post-process its output to retrieve better segmentations and alignments. The basic units in English and Chinese are respectively single letters and single Chinese characters in the M2M-Aligner's input. For English, some letter combinations, namely *ch, ck, sh* and two identical letters appearing next to each other are always pronounced as single letters and hence never aligned to different Chinese characters. We pre-contract them so that the M2M-Aligner will treat them as single letters and never segment those combinations incorrectly.

Due to the fact that single Chinese characters are normally independent transliteration units, in most cases several English letters are aligned to one Chinese character. The letter *x* is the only exception as it may be aligned to two Chinese characters, which will be handled by post-processing in this paper. Despite of that, we set the maximum length of substring on the English side as six and on the Chinese side as one. All the other parameters of the M2M-Aligner have default settings.

Table 1 shows an output sample. In order to reduce the segmentation and alignment errors further, we first modify the alignments associated with *x* and then post-process the output using a hard-constrained EM algorithm.

It is easy to find from the training data that when the letter *x* should be mapped to two neighboring characters A and B, A's corresponding pinyin is always *ke* and B's pinyin always starts with *s* or *x*. With the help of pinyin, it is straightforward to extract all the instances in which *x* should be aligned to two Chinese characters but have been incorrectly processed by the M2M-Aligner. For those instances, we erase the boundaries between the two Chinese characters A, B which *x* is aligned to. On the English side, we remove the boundary closest to *x*. After the modification, the third and fourth instances in Table 1 will be changed as the ones in Table 2. The segmentations and alignments are still not correct but it is now possible to continue with the next stage.

a|ber|nat|hy|	阿|伯|内|西|
a|ber|ne|thy|	阿|伯|内|西|
t|e|xi|do|	特|克|西|多|
wi|ll|c|o|x|	威|尔|科|克|斯|

Table 1: Sample output of M2M Aligner

t|exi|do|	特|克西|多|
wi|ll|c|ox|	威|尔|科|克斯|

Table 2: Sample segmentations and alignments

We assume that the segmentations and alignments with low frequencies are very likely to be errors produces by the M2M-Aligner. In this respect, we develop an algorithm which largely reduces the low frequency terms and therefore significantly improves the segmentation and alignment quality. Given the current output, we estimate the probability of an individual instance s by:

$$p(s) = \prod_{i=1}^{n} p(e_i) p(e_i \leftrightarrow c_i) \tag{1}$$

where $p(e_i)$ is the probability of segmented substring e_i on the English side and $p(e_i \leftrightarrow c_i)$ is the probability of e_i aligned to c_i, which is on the Chinese side. Using maximum likelihood estimation (MLE), $p(e_i)$ and $p(e_i \leftrightarrow c_i)$ are calculated as:

$$p(e_i) = \frac{c(e_i) + 1}{N + R} \tag{2}$$

$$p(e_i \leftrightarrow c_i) = \frac{c(e_i \leftrightarrow c_i) + 1}{N + R} \tag{3}$$

N is the total number of segmented substrings or alignments. R is the number of unique substrings, which works as a smoothing factor. $c(e_i)$ and

$c(e_i \leftrightarrow c_i)$ are respectively the counts of the substring e_i and corresponding alignment.

We use the obtained probabilities to reassess and modify the current segmentations and alignments. To maximize the probability presented in formula 1, a local greedy search strategy is used for efficiency. For every two neighboring substrings on the English side, we find the best split point as their new boundary. The probabilities are updated afterwards. This procedure iterates until it converges. Table 3 shows the segmentation and alignment results after the EM post-processing. According to error inspection, the refined result is significantly better than the original one even though there are still mistakes involved.

<table>
<tr><td>a|ber|na|thy|</td><td>阿|伯|内|西|</td></tr>
<tr><td>a|ber|ne|thy|</td><td>阿|伯|内|西|</td></tr>
<tr><td>t|exi|do|</td><td>特|克西|多|</td></tr>
<tr><td>wi|ll|co|x|</td><td>威|尔|科|克斯|</td></tr>
</table>

Table 3: Sample segmentations and alignments

3.2 Phrase-Based Machine Transliteration

In this paper, we build our phrase-based transliteration systems with Moses using the refined outputs of the M2M-Aligner. The output can be easily converted into the format of alignment files that are generated by Moses after its third training step. We build the system from step four with default parameters. We use IRSTLM (Federico et al., 2008) to build language models with order 6.

For English to Chinese transliteration, we build two systems with different transliteration units on the English side. First, we build a full character based system, in which all the single letters are basic mapping units. At the decoding stage, the source English names can be input directly as strings of letters and the Moses decoder will identify the phrase boundaries and map the phrases as transliteration units to target Chinese characters.

Additionally, we build a system with presegmented substrings on the English side as basic units. In this case, at the decoding stage, pre-segmenting the source English names is required. A CRF segmentation model is trained using the CRF++ toolkit. However, since the CRF model essentially does the segmentation via identifying the boundaries, some produced substrings are not known to the transliteration model. They are treated as OOVs and therefore will not be transliterated. Under these circumstances, we combine the two systems. When the input cannot be transliterated by the system built with presegmented substrings, the output of the character based system is used as backoff.

For Chinese to English transliteration, we build two character based systems. The first one is trained with Chinese characters and the second one with corresponding Chinese pinyin. The pinyin based system is used similarly as backoff because occasionally there are some uncommon Chinese characters that are not seen in the training data. However, there are always Chinese characters contained in the training data that share the same pronunciations as the unknown ones. They also have the same pinyin as it is the phonetic representation of Chinese character.

All the systems are tuned with the official development data sets.

3.3 Using Multilingual Resources

Transliteration is based on phonetics and therefore it is heavily language dependent. The western names associated with transliteration tasks are written in the same script but actually have different language origins. Thus, they should be transliterated using different language specific systems.

We use the dictionary *Chinese Transliteration of Foreign Personal Names* (Xia, 1993) as our bilingual resources, which is also used in Li et al. (2004)'s research. It contains western names from different language origins and their Chinese transliterations. In this research, we choose the western language sources that have more than 10,000 terms in the dictionary to build backoff transliteration systems introduced in the previous section. The chosen languages are Czech, English, Finnish, French, Turkish, German, Portuguese, Hungarian, Italian, Romanian, Russian, Spanish, Swedish and Serbian.

For the English to Chinese development set, 1,783 instances out of 2,802 are found in the dictionary. Among them, 1,645 have at least one correct transliteration in the dictionary while 318 have at least one correct transliteration that is not in the dictionary. The statistics is similar for the Chinese to English development set.

For the test data, we apply the source name to all the language specific systems. For each term, every system returns 10 different scores of Moses, such as total score, language model score, phrase

		Standard Runs					Non-Standard Runs				
Tasks	Test Sets	Configuration	ACC	F-score	MRR	MAP_{ref}	Configuration	ACC	F-score	MRR	MAP_{ref}
EnCh	NEWS11	Character Based	0.324	0.682	0.404	0.312	Baseline	0.365	0.708	0.431	0.351
		Subtring Based	0.333	0.673	0.387	0.320	Reranking	0.722	0.870	0.775	0.717
		Backoff System	0.340	0.694	0.397	0.327					
	NEWS12	Character Based	0.311	0.660	0.396	0.303	Baseline	0.373	0.693	0.436	0.363
		Subtring Based	0.325	0.660	0.384	0.313	Reranking	0.656	0.824	0.735	0.649
		Backoff System	0.335	0.676	0.396	0.323					
ChEn	NEWS11	Character Based	0.150	0.755	0.228	0.150	Baseline	0.165	0.773	0.252	0.164
		Pinyin Based	0.109	0.731	0.183	0.109	Reranking	0.354	0.833	0.428	0.354
		Backoff System	0.153	0.768	0.233	0.153					
	NEWS12	Character Based	0.191	0.711	0.271	0.187	Baseline	0.214	0.745	0.305	0.212
		Pinyin Based	0.146	0.712	0.223	0.143	Reranking	0.345	0.805	0.421	0.345
		Backoff System	0.199	0.752	0.280	0.194					

Table 4: Official Results

score and different translation model scores. They can be used as features for reranking these outputs by different systems. With respect to the mean F-score, we train a linear regression model using WEKA (Hall et al., 2009) on the development data sets and use it as the reranking system. Additionally, the baseline systems are trained only using the English data from the dictionary to be compared with the multilingual reranking model.

4 Experimental Results and Analysis

Table 4 shows the official experimental results.

4.1 Standard Runs

Since the test data sets are the same as the ones used in the NEWS transliteration shared tasks of 2011 and 2012, our systems are compared to the evaluated systems in the previous years.

For English to Chinese, our system beats all the systems of 2012 (Zhang et al., 2012) but fails to beat the best performing system of 2011 (Zhang et al., 2011) according to ACC. Generally, the substring based system achieves better results than the character based system, which indicates that the CRF model is more effective in identifying phrase boundaries than Moses.

For Chinese to English, our system is slightly worse than the best performing systems but still very competitive. We can see that the Chinese character based system yields better results. Compared to pinyin, Chinese characters contain more information that is useful to transliteration.

As expected, the backoff systems perform best in both tasks. It is also notable that our systems perform better on NEWS12 test data sets, probably because the NEWS12 test data are more similar to the development sets that are used for tuning.

4.2 Non-Standard Runs

For both tasks, our multilingual reranking models significantly outperform the baseline systems. We saw earlier that the dictionary used for training covers a substantial part of the development sets and we assume it is similar for the test sets. Nevertheless, adding multilingual resources leads machine transliteration quality to a new level.

Transliteration without language source discrimination is very difficult because the phonetic systems of different languages are very inconsistent. Take an instance from the development data, *Arbos* as a Spanish name is transliterated as 阿沃斯 in Chinese. If using the transliteration system trained with English names, it is almost impossible to obtain the correct transliteration because *b* is never pronounced as *v* in English.

Our multilingual reranking model can be improved further via adding more multilingual resources, using more effective features for reranking and adopting better regression algorithms.

5 Conclusions

We build phrase based transliteration systems using Moses with refined alignments of the M2M-Aligner. The evaluation results of the standard runs indicate that our approaches are effective in solving both English to Chinese and Chinese to English transliteration tasks. The results of the non-standard runs demonstrate that the transliteration quality can be greatly improved using multilingual resources and good reranking techniques.

References

Yu Chen, Rui Wang, and Yi Zhang. 2011. Statistical machine transliteration with multi-to-multi joint source channel model. In *Proceedings of the Named Entities Workshop Shared Task on Machine Transliteration*, Chiang Mai, Thailand.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. Irstlm: an open source toolkit for handling large scale language models. In *INTERSPEECH*, pages 1618–1621. ISCA.

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*, NEWS '10, pages 48–52, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

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, April. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chan-Hung Kuo, Shih-Hung Liu, Tian-Jian Mike Jiang, Cheng-Wei Lee, and Wen-Lian Hsu, 2012. *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, chapter Cost-benefit Analysis of Two-Stage Conditional Random Fields based English-to-Chinese Machine Transliteration, pages 76–80. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ying Qin and GuoHua Chen, 2011. *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, chapter Forward-backward Machine Transliteration between English and Chinese Based on Combined CRFs, pages 82–85. Asian Federation of Natural Language Processing.

Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string edit distance. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(5):522–532, May.

Defu Xia. 1993. *Translation Dictionary for Foreign Names*. China Translation and Publishing Corporation, Beijing, China, October.

Min Zhang, Haizhou Li, A. Kumaran, and Ming Liu. 2011. Report of news 2011 machine transliteration shared task. In *Proceedings of the 3th Named Entity Workshop*, NEWS '11, pages 10–20, Stroudsburg, PA, USA. Association for Computational Linguistics.

Min Zhang, Haizhou Li, A. Kumaran, and Ming Liu. 2012. Report of news 2012 machine transliteration shared task. In *Proceedings of the 4th Named Entity Workshop*, NEWS '12, pages 10–20, Stroudsburg, PA, USA. Association for Computational Linguistics.

Min Zhang, Haizhou Li, A. Kumaranz, and Rafael E. Banchs. 2015. Whitepaper of NEWS 2015 shared task on transliteration generation. In *NEWS '15 Proceedings of the 2015 Named Entities Workshop: Shared Task on Transliteration*, Beijing, China.