Forward-backward Machine Transliteration between English and Chinese Based on Combined CRFs

Ying Qin
Department of Computer Science,
Beijing Foreign Studies University
qinying@bfsu.edu.cn

Guohua Chen
National Research Centre for Foreign Language Education, Beijing Foreign Studies University
professorchenguohua@yahoo.com.cn

Abstract

The paper proposes a forward-backward transliteration system between English and Chinese for the shared task of NEWS2011. Combined recognizers based on Conditional Random Fields (CRF) are applied to transliterating between source and target languages. Huge amounts of features and long training time are the motivations for decomposing the task into several recognizers. To prepare the training data, segmentation and alignment are carried out in terms of not only syllables and single Chinese characters, as was the case previously, but also phoneme strings and corresponding character strings. For transliterating from English to Chinese, our combined system achieved Accuracy in Top-1 0.312, compared with the best performance in NEWS2011, which was 0.348. For backward transliteration, our system achieved top-1 accuracy 0.167, which is better than others in NEWS2011.

1 Introduction

The surge of new named entities is a great challenge for machine translation, cross-language IR, cross-language IE and so on. Transliteration, mostly used for translating personal and location names, is a way of translating source names into target language with approximate phonetic equivalents (Li et al., 2004), while backward transliteration traces back to the foreign names (Guo and Wang, 2004). Phonetic-based and spelling-based approaches are popularly applied in machine transliteration (Karimi et al. 2011). Recently direct orthographical mapping (DOM) between two languages, a kind of spelling-based transliteration approach, outperforms that of phonetic-based methods. Most systems in NEWS2009 and NEWS2010 utilized this approach to automatic transliteration (Li et al., 2009; Li et al., 2010).

In previous researches, syllable segmentation and alignment were done in terms of single syllables in training a transliteration model. (Yang et al., 2009; Yang et al., 2010; Aramaki and Abekawwa, 2009; Li et al., 2004). Sometimes, however, it is hard to split an English word and align each component with a single Chinese character, which is always monosyllabic. For instance, when *TAX* is transliterated into 塔克斯 (Ta Ke Si) in Chinese, no syllable is mapped onto the characters 克 and 斯, for *X* is pronounced as two phonemes rather than a syllable. In this paper, we try to do syllable segmentation and alignment on a larger unit, that is, phoneme strings.

Conditional Random Fields (CRF) was successfully applied in transliteration of NEWS2009 and NEWS2010 (Li et al. 2009; Li et al. 2010). Transliteration was viewed as a task of two-stage labeling (Yang et al. 2009; Yang et al., 2010; Aramaki and Abekawwa, 2009). Syllable segmentation was done at the first stage, and then target strings were assigned to each chunk at the next stage. The huge amounts of features in the second stage made model training time-consuming. Thirteen hours on an 8-core server were expended to train the CRF model in the work done by Yang et al. (2010).

To reduce training time and requirement of high-specification hardware, we adopt a combined CRF transliteration system by dividing the training data into several pools and each being used to train a recognizer to predict the target characters. The final transliteration results are the arranged according to the probabilities of all CRF outputs.

In the following, section 2 describes how segmentation and alignment are done on the unit of phoneme strings. Section 3 explains how the forward-backward transliteration system between English and Chinese is built. Performances of the

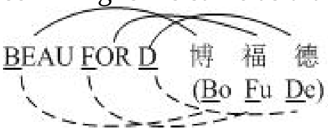
Proceedings of the 2011 Named Entities Workshop, IJCNLP 2011, pages 82–85,
Chiang Mai, Thailand, November 12, 2011.

system on all the metrics of NEWS2011 are listed in section 4, which is followed by discussions. The last section is the conclusion.

2 Segmentation and Alignment

Lack of gold standard syllable segmentation and alignment data is an obstacle to transliteration model training. Yang et al. (2009) applied N-gram joint source-channel and EM algorithm, while Aramaki and Abekawwa (2009) made use of word alignment tool in GIZA++ to obtain a syllable segmentation and alignment corpus from the training data given. Neither of them reported how precise their alignments were. Yang et al. (2010) proposed a joint optimization method to reduce the propaganda of alignment error.

Pinyin is known as romanized pronunciation of Chinese characters. Due to the nature of *pinyin*, there are many similarities between English orthography and Chinese *pinyin*. Of the 24 English consonants, 17 have almost the same pronunciation in *pinyin*. Since English orthography has a close relationship with phonetic symbols, we believe that consonants in *pinyin* can also provide clues for syllable segmentation and alignment. In the following example, the consonant sequence in English is same as that in *pinyin*.



Therefore we can do syllable segmentation with the help of pronunciations of Chinese characters. Segmentation is carried out from the second character, for there is no need to split from the initial letter of a string.

However not all mappings between spelling and phoneme are involved in this approach. The following cases are insolvable.

Case 1: there is no corresponding consonant. For instance, ARAD 阿拉德 (A La De).

Case 2: several letters occupy one phoneme. For instance, BAECK 贝克 (Bei Ke).

Case 3: duplicate letters cause ambiguity. For instance, ANNADA LE 安娜代尔 (An Na Dai Er).

Case 4: consonants are sometimes mismatched. For instance, ACQUARELLI 阿奎雷利 (A Kui Lei Li).

Case 5: there are inconsistencies complicating the situation. For instance: ADDINGTON 阿丁顿 (A Ding Dun).

Therefore *pinyin*-based segmentation is only treated as a preliminary result.

To deal with case 1, we take a two-step matching—strict matching and then loose matching—between the consonant in *pinyin* and the English word. If the same consonant is not available, strings of a similar pronunciation are sought. For instance, the consonant in *pinyin* Fu is *f*, if there is no letter *f* in the English transliteration, *v, ph, gh* are adopted for segmentation.

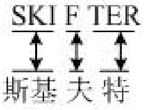
We apply transformation rules to optimize the syllable alignment result. The rules are induced manually by observation of segmentation errors. We believe gold alignment training corpora are the foundation of good performance no matter which algorithms is applied.

However, we find that some chunks in English correspond to Chinese strings in most translations. Some of such chunks are given in Table 1 as examples. We keep the alignment between these chunks and corresponding Chinese character strings, calling it phoneme strings based alignment.

SKIN 斯金	SKI 斯基	SCO 斯科
MACA 麦考	MACA 麦卡	MACC 麦克
MACKI 麦金	X 克斯	SKEW 斯丘

Table 1. Alignment of English chunks and corresponding Chinese character strings

The alignment of phoneme strings has advantages over single phoneme alignment. Since each English syllable string may be mapped onto several possible Chinese characters, there will be fewer choices if the alignment is based on phoneme strings when an English syllable sequence is finally transliterated into Chinese character strings. For example, *s* can be mapped onto the Chinese characters 斯(Si), 丝(Si) and 思(Si), *ky* can be mapped onto 基(Ji), 吉(Ji) and 季(Ji), but for *sky*, it is usually transliterated into 斯基(Si Ji), not others sequences serve as alternatives. Therefore, we think phoneme strings alignment is better than single phoneme alignment. The following is an example of alignment based on phonemes strings.



As to the backward transliteration, segmentation and alignment are also based on phoneme strings. Following are two columns of aligned data for CRF model training.

哈	HA
克斯	X

3 Forward and Backward Transliteration System

CRF is a discriminative model and makes a global optimum prediction according to the conditional probability (Lafferty et al., 2001). When applying CRF to transliteration, the task is treated as labeling source words with target language strings. Similar to previous works (Yang et al., 2010; Aramaki and Abekawwa, 2009), we build a two-stage CRF transliteration system between English and Chinese. The first stage CRF decoder splits the source words into several chunks. Outputs of the first stage are then sent to the second CRF to label what target characters are transliterated. The final transliteration of the source word is the sequence of all the target characters.

For training the CRF chunker with the given corpora segmented and aligned, each character is labeled with the *BI* scheme, that is, *B* for the beginning character of a chunk, *I* for the characters in other position. For example, in English to Chinese training data, *ABBE* is segmented and aligned as follows.

A	阿
BBE	贝

The two-column data for training the CRF chunker is,

A	*B*
B	*B*
B	*I*
E	*I*

The window size is set as 3, the same as the experiment by Aramaki and Abekawwa (2009).

Though a larger window is propitious to provide more contextual information, there are too many features for training the second stage CRF. We have to reduce the window size. In the second stage of CRF training, the window size is 2, that is, features used are C_{-2}, C_{-1}, C_0, C_1, C_2, $C_{-1}C_0$, C_0C_1, $C_{-2}C_{-1}C_0$ and $C_0C_1C_2$, which C_0 denotes the current chunk. Still the time it takes to train a model on a normal PC is intolerably long[1].

Even the training data aligned on phoneme strings are checked manually, errors are still sometimes somewhere. To reduce the risk of local errors in segmentation and alignment, we divide the training data randomly and evenly into several pools. The size of the pools is set simply according to the capability of our PCs. If some errors occur in some pools but not in all, a correct predication can still be made by the CRFs trained on correct pools.

The combined CRF recognizers are both used for forward and backward transliterations at the second stage. The workflow of our transliteration system is depicted in Figure 1.

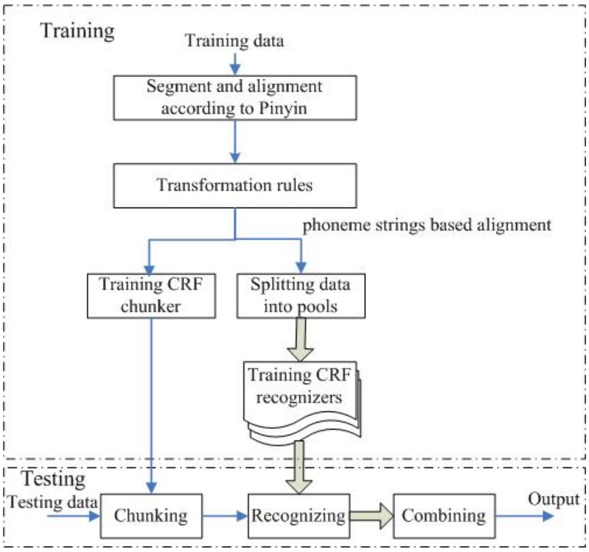


Figure 1. Workflow of Transliteration System

4 Performances and Discussion

We use the open CRF++[2] toolkits to build the two-stage CRF transliteration with all given data of NEWS2011.

4.1 Performances

The number of recognizers may affect the performance of the whole system. To suit the best capacity of our PC, we train 10 forward and 20 backward recognizers. We also train another forward transliteration consisting of 20 recognizers for comparison. Due to time limit, we do not try other numbers in backward and forward transliteration during NEWS2011. Because the test data of NEWS2011 are reserved for future use, we can not try other numbers to build transliteration systems for comparison.

Table 2 shows the common evaluation of our transliteration system between English (E) and Chinese (C). We can see that the performance of E->C transliteration varies slightly with different numbers of combination on all evaluation metrics. The performance of backward transliteration is lower than that of the forward direction on ACC but is better on Mean F score.

[1] Using the same parameters setting of CRF learner as Aramaki and Abekawwa (2009), the training time on a PC (2.3GHZ, 4GB) with NEWS2011 data (37753 English names) reaches 4800 hours.

[2] http://crfpp.sourceforge.net/

	CRFs	ACC	Mean F	MRR	MAP_{ref}
E->C	10	0.312	0.669	0.339	0.310
	20	0.308	0.666	0.337	0.306
C->E	20	0.167	0.765	0.202	0.167

Table 2. Performance of Combined Transliteration System

4.2 Discussions

- Granularity of syllable segmentation and alignment

Preprocessing training data on phoneme strings alignment is our approach in attempting to improve transliteration between English and Chinese. In backward transliteration, our system is better than others in the shared task of NEWS2011. Can we assume that larger granularity alignment is better than a smaller one? Which granularity is optimum?

- Number of CRF recognizer

With more data, the time it takes to train a model based on CRF increases sharply. We train transliteration models with the same algorithm but different usage of data and then combine the results of all recognizers. In this way, training time is reduced. However we can see from the result of testing that the performance of transliteration varies with the number of recognizers. What is the comparison between combined system and single system? Which number of combinations is the best? We will need to explore these questions with more data.

5 Conclusion

Two-stage CRFs are applied to transliterating between English and Chinese. We try to improve the performance from two directions, one is training data processing, which is segmented and aligned based on phoneme strings; another is system building, in which several models on different parts of data are trained and their outputs are combined. The final results of the transliteration are arranged in sequential order in accordance with the degree of probability of all the recognizers.

In future work, we will focus on good standard data and methods of combination to further improve the performance of forward-backward transliteration system.

Acknowledgments

The work is supported by the National Social Science Fund (No. 10CYY024). Thanks are due to the support of National Research Centre for Foreign Language Education, BFSU.

We are grateful to all anonymous reviewers.

References

Eiji Aramaki and Takeshi Abekawa. 2009. Fast decoding and easy implementation: Transliteration as a sequential labeling. *Proceeding of ACL/IJCNLP*. Named Entities Workshop Shared Task. 65-68.

Yuqing Guo, Haifeng Wang. 2004. Chinese-to-English Backward Machine Transliteration. Companion Volume to *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP 04)*. 17-20.

Sarvnaz Karimi, Falk Scholer and Andrew Turpin. 2011. *Machine Transliteration Survey*. ACM Computing Surveys, 43(4): 1–57.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning (ICML01).*

Chun-Jen Lee and Jason S. Chang. 2003. Acquisition of English-Chinese Transliterated Word Pairs from Parallel-Aligned Texts Using a Statistical Machine Transliteration Model. *Proceedings of HLT-NAACL*. 96-103.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. *Proceedings of 42nd ACL Annual Meeting*. 159–166.

Haizhou Li, A Kumaran, Vladimir Pervouchine and Min Zhang. 2009. Report of NEWS 2009 Machine Transliteration Shared Task. *Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP*. 1–18.

Haizhou Li, A Kumaran, Min Zhang and Vladimir Pervouchine. 2010. Report of NEWS 2010 Transliteration Generation Shared Task. *Proceedings of the 2010 Named Entities Workshop, ACL 2010.* 1–11.

Dong Yang, Paul Dixon, Yi-Cheng Pan, Tasuku Oonishi, Masanobu Nakamura and Sadaoki Furui. 2009. Combining a Two-step Conditional Random Field Model and a Joint Source Channel Model for Machine Transliteration. *Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP*. 72–75.

Dong Yang, Paul Dixon and Sadaoki Furui. 2010. Jointly optimizing a two-step conditional random field model for machine transliteration and its fast decoding algorithm. *Proceedings of the ACL 2010. Conference Short Papers*. 275–280.