Syllable-based Machine Transliteration with Extra Phrase Features

Chunyue Zhang, Tingting Li, Tiejun Zhao
MOE-MS Key Laboratory of Natural Language Processing and Speech
Harbin Institute of Technology
Harbin,China
{cyzhang,ttli,tjzhao}@mtlab.hit.edu.cn

Abstract

This paper describes our syllable-based phrase transliteration system for the NEWS 2012 shared task on English-Chinese track and its back. Grapheme-based Transliteration maps the character(s) in the source side to the target character(s) directly. However, character-based segmentation on English side will cause ambiguity in alignment step. In this paper we utilize Phrase-based model to solve machine transliteration with the mapping between Chinese characters and English syllables rather than English characters. Two heuristic rule-based syllable segmentation algorithms are applied. This transliteration model also incorporates three phonetic features to enhance discriminative ability for phrase. The primary system achieved 0.330 on Chinese-English and 0.177 on English-Chinese in terms of top-1 accuracy.

1 Introduction

Machine transliteration, based on the pronunciation, transforms the script of a word from a source language to a target language automatically.

With a continuous growth of out-of-vocabulary names to be transliterated, the traditional dictionary-based methods are no longer suitable. So data-driven method is gradually prevailing now, and many new approaches are explored.

Knight(1998) proposes a phoneme-based approach to solve the transliteration between English names and Japanese katakana. It makes use of a common phonetic representation as a pivot.

The phoneme-based approach needs a pronunciation dictionary for one or two languages. These dictionaries usually do not exist or can't cover all the names. So grapheme-based(Li et al., 2004) approach has gained lots of attention recently. Huang(2011) proposes a novel nonparametric Bayesian using synchronous adaptor grammars to model the grapheme-based transliteration. Zhang(2010) builds the pivot transliteration model with grapheme-based method.

The hybrid approach tries to utilize both phoneme and grapheme information, and usually integrates the output of multiple engines to improve transliteration. Oh and Choi(2006) integrate both phoneme and grapheme features into a single leaning framework.

As an instance of grapheme-based approach, Jia(2009) views machine transliteration as a special example of machine translation and uses the phrase-based machine translation model to solve it. The approach is simple and effective. Our paper follows this way. However, using the English letters and Chinese characters as basic mapping units will make ambiguity in the alignment and translation step. One Chinese character usually maps one syllable, so syllabifying English words can be more discriminative.

We present a solution to this ambiguity by replacing the English character with an English syllable which is consecutive characters and can keep some phonetic properties. For this purpose, two heuristic and simple syllable segmentation algorithms are used to syllabify English side into syllables sequence. Besides two above, three extra phrase features for transliteration are used to enhance the model.

The rest of this paper is organized as follows. Section 2 introduces the phrase-based model briefly. Section 3 describes two rule-based syllable

segmentation methods and three new special features for transliteration in detail. Experiments and analyses are discussed in section 4. Conclusions and future work are addressed in section 5.

2 Phrase-based Machine Transliteration Model

Machine transliteration can be regarded as a special instance of machine translation. Jia(2009) solves transliteration with phrase-based model firstly. There an English character is treated as a word in machine translation. On the contrast, character is replaced by syllable in this paper. Then transliteration can be viewed as a pure translation task. The phrase-based machine transliteration can be formulated by equation 1.

$$\tilde{e} = \arg\max_{e} p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x) \qquad (1)$$

- n is the number of features
- λ_i is the weight of feature i

In our phrase-based transliteration system, the following features are used by default:

- the bidirectional probability between source phrase and the target phrase
- The bidirectional lexical probability between source phrase and target phrase
- the fluency of the output, namely language model
- the length penalty

3 Syllable Segmentation and Extra Phrase Features

This section describes two rule-based syllable segmentation algorithms and three extra phrase features added to machine transliteration model.

3.1 Syllable Segmentation Algorithm

In (Jia et al., 2009), the basic alignment units are English character and Chinese character(called c2c). This setup is the simplest format to implement the model. However, transliteration from English to Chinese usually maps an English syllable to a single Chinese character. As one Chinese character usually corresponds to many English characters, the c2c method has only a modest discriminative ability. Obviously syllabifying English is more suitable for this situation. Yang(2010) utilizes a CRF-based segmentor to syllabify English and Kwong(2011) syllabifies English with the Onset First Principle. Alternatively, inspired by (Jiang, 2007), two heuristic rule-based methods are addressed to syllabify the English names in this paper.

Given an English name *E*, it can be syllabified into a syllable sequence *SE* = {*e1*,*e2*,...,*en*} with one of the following two linguistic methods.

Simple Segmentation Algorithm(SSA):

1. {'*a*', '*o*' , '*e*', '*i*', '*u*'} are defined as vowels. '*y*' is defined as a vowel when it is not followed by a vowel; '*r*' is defined as a vowel when it follows a vowel and is followed by a consonant[1]. All other characters are defined as consonants; this forms the basic vowel set;
2. A consecutive vowels sequence, formed by the basic vowel set, is treated as a new vowel symbol; Step 1 and 2 form the new vowel set;
3. A consonant and its following vowel are treated as a syllable;
4. Consecutive consonants are separated; a vowel symbol(in the new vowel set) followed by a consonant is separated;
5. The rest isolated characters sequences are regarded as individual syllables in each word.

SSA treats all the consecutive vowels as a single new vowel simply. In fact, many consecutive vowels like "*io*" often align two or more Chinese characters, such as " *zio* 西 奥". It is better to separate it as two syllables rather than one syllable in alignment step. So we present another segment algorithm which takes more details into consideration.

Fine-grained Segment Algorithm(FSA):

1. Replace '*x*' in English names with '*k s*' firstly;
2. {'*a*','*o*','*e*','*i*','*u*'} are defined as vowels. '*y*' is defined as a vowel when it is not followed by a vowel;
3. When '*w*' follows '*a*','*e*','*o*' and isn't followed by '*h*', treat '*w*' and the preceding vowel as a new vowel symbol; Step 2 and 3 form the basic vowel set;
4. A consecutive vowels sequence which is formed by the basic vowel set is treated as a new vowel

[1] A review points the SSA lacking of ability to deal with '*h*'. We leave it for the future work.

symbol, excepting '*iu*', '*eo*', '*io*', '*oi*', '*ia*', '*ui*', '*ua*', '*uo*' ; Step 2, 3 and 4 form the new vowel set;
5. Consecutive consonants are separated; a vowel symbol(in the new vowel set) followed by a consonant sequence is separated;
6. A consonant and its following vowel are treated as a syllable; the rest of the isolated consonants and vowels are regarded as individual syllables in each word.

After segmenting the English characters sequence, the new transliteration units, syllables, will be more discriminative.

3.2 Extra phrase features

The default features of phrase can't express the special characteristic of transliteration. We propose three features trying to explore the transliteration property.

Begin and End Feature(BE)

When a Chinese character is chosen as the corresponding transliteration, its position in the transliteration result is important. Such as a syllable "*zu*" that can be transliterate into "朱" or "祖" in Chinese while "朱" will be preferred if it appears at the beginning position.

To explore this kind of information, the pseudo characters "B" and "E" are added to the train and test data. So in the extracted phrase table, "B" always precedes the Chinese character that prefers at the first position, and "E" always follows the Chinese character that appears at the last position.

Phrase Length Feature

Chinese character can be pronounced according to its pinyin format which is written like English word. And the longer English syllable is, the longer pinyin format it often has. So the length information of Chinese character and its pinyin can be used to disambiguate the phrase itself. Here we definite two new features to address it. Suppose <e,c> as a phrase pair, e= {e1,e2,...,em},c = {c1,c2,...,cn},ei stands for an English syllable and ci stands for a Chinese character. p(ci) is the pinyin format of ci. #(ei) is equal to the number of characters in a syllable. #p(cj) is equal to the number of characters in a pinyin sequence. And then,

$$L1 = Sum(\#(ei)) / Sum(\#(p(cj))$$
$$L2 = m / n$$

4 Experiments

This section describes the data sets, experimental setup, experimental results and analyses.

4.1 Data Sets

The training set of English-Chinese transliteration track contains 37753 pairs of names. We pick up 3000 pairs from the training data randomly as the closed test set and the rest 34753 pairs as our training data set. In the official dev set some semantic translation pairs are found, such as "REPUBLIC OF CUBA 古巴共和国", and some many-to-one cases like "SHELL BEACH 谢尔比奇" also appear. We modify or delete these cases from the original dev set. At last, 3223 pairs are treated as the final dev set to tune the weights of system features.

Language	Segmentation Algorithm	Number
English	Character-based	6.82
	SSA	4.24
	FSA	4.48
Chinese	Character-based	3.17

Table 1: Average syllables of names based on different segmentation methods

Language	Segmentation Algorithm	Number
English	Character-based	26
	SSA	922
	FSA	463
Chinese	Character-based	368

Table 2 :Total number of unique units

For the Chinese-English back transliteration track, the final training and test sets are formed in the same way; the original dev set is used directly.

Here we use Character-based which treats single character as a "syllable", Simple and Fine-grained segmentation algorithms to deal with English names. Table 1 and table 2 show some syllabic statistics information. Table 1 shows the average syllables of the three segmentation approaches in training data. Table 2 shows the total number of unique units.

4.2 Experimental Setup

The Moses (Koehn et al., 2007) is used to implement the model in this paper. The Srilm(Stolcke et al., 2002) toolkit is used to count

n-gram on the target of the training set. Here we use a 3-gram language model. In the transliteration model training step, the Giza++(Och et al., 2003) generates the alignment with the grow-diag-and-final heuristic, while other setup is default. In order to guarantee monotone decoding, the distortion distance is limited to 0. The MERT is used to tune model's weights. The method of (Jia et al., 2009) is the baseline setup.

4.3 Evaluation Metrics

The following 4 metrics are used to measure the quality of the transliteration results (Li et al., 2009a): Word Accuracy in Top-1 (ACC), Fuzziness in Top-1 (Mean F-score), Mean Reciprocal Rank (MRR), MAPref.

4.4 Results

Table 3 shows the performance of our system corresponding to baseline, SSA and FSA on the closed test set of EnCh track. BE, L1,L2 and BE+L1+L2 are implemented on the basis of FSA.

	ACC	Mean F-score	MRR	MAPref
Baseline	0.628	0.847	0.731	0.628
SSA	0.639	0.850	0.738	0.639
FSA	0.661	0.861	0.756	0.661
BE	0.648	0.856	0.751	0.648
L1	0.661	0.864	0.756	0.661
L2	0.619	0.844	0.727	0.619
BE+L1+L2	0.665	0.863	0.762	0.665

Table 3:The held-in results of EnCh

Table 3 shows that the forward transliteration performance gets consistent improvement from baseline to FSA. None of new three features can improve by self, while combining three features can gain a little.

	ACC	Mean F-score	MRR	MAPref
EnCh_Pri	0.330	0.676	0.408	0.319
EnCh_2	0.317	0.667	0.399	0.308
ChEn_pri	0.177	0.702	0.257	0.173

Table 4: The final official results of EnCh and ChEn

According to the performance of closed test, the transliteration results of EnCh and ChEn based on BE+L1+L2 are chosen as the primary submissions(EnCh_Pri and ChEn_Pri). And the result of FSA is the contrastive submission(EnCh_2). The table 4 shows the final official results of EnCh and ChEn.

5 Conclusions and future work

This paper uses the phrase-based machine translation to model the transliteration task and the state-of-the-art translation system Moses is used to implement it. We participate in the NEWS 2012 Machine Transliteration Shared Task English-Chinese and Chinese-English tracks.

To improve the capability of the basic phrase-based machine transliteration, two heuristic and rule-based English syllable segmentation methods are addressed. System can also be more robust with combination of three new special features for transliteration. The experimental results show that the Fine-grained Segmentation can improve the performance remarkably in English-Chinese transliteration track.

In the future, extensive error analyses will be made and methods will be proposed according to the specific error type. More syllable segmentation methods such as statistical-based will be tried.

Acknowledgments

The authors would like to thank all the reviews for their help about correcting grammatical errors of this paper and invaluable suggestions. This work is supported by the project of National High Technology Research and Development Program of China (863 Program) (No. 2011AA01A207) and the project of National Natural Science Foundation of China (No. 61100093).

References

Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In Proc. of ICSLP, Denver, USA.

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. In Proceedings of the ACL 2010 Conference Short Papers. pp. 275--280 Uppsala, Sweden.

Franz Josef Och, Hermann Ney. 2003. A systematic comparison of various statistical alignment models. Comput.Linguistics 29, 1, 19–51.

Haizhou Li , Min Zhang, Jian Su. 2004. A Joint Source Channel Model for Machine Transliteration. In Proceedings of the 42nd ACL, pp. 159-166.

Kevin Knight, Jonathan Graehl. 1998. Machine Transliteration. Computational Linguistics, Vol. 24, No. 4, pp. 599-612.

Long Jiang , Ming Zhou , Leefeng Chien and Cheng Niu. Named entity translation with web mining and transliteration, Proceedings of the 20th international joint conference on Artifical intelligence, p.1629-1634, January 06-12, 2007, Hyderabad, India

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.

Oi Yee Kwong. 2011. English-Chinese Personal Name Transliteration by Syllable-Based Maximum Matching. In the Proceedings of the 2011 Named Entities Workshop,2011,pp.96-100.

Philipp Koehn, Hieu Hoang, Marcello Federico Nicola Bertoldi , Brooke Cowan and Wade Shen . 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the 45th ACL Companion Volume of the Demo and Poster Sessions, pp. 177-180.

Yun Huang, Min Zhang and Chewlim Tan. 2011. Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars. In Proceedings of ACL-HLT 2011: Short Papers,Portland, Oregon, pp.534-539.

Yuxiang Jia, Danqing Zhu, and Shiwen Y. 2009. A Noisy Channel Model for Grapheme-based Machine Transliteration, In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 88-91.