A Syllable-based Name Transliteration System

Xue Jiang [1, 2]
[1]Institute of Software, Chinese Academy of Science.
Beijing China, 100190
jiangxue1024@yahoo.com.cn

Le Sun [1], Dakun Zhang [1]
[2]School of Software Engineering, Huazhong University of Science and Technology. Wuhan China, 430074
sunle@iscas.ac.cn
dakun04@iscas.ac.cn

Abstract

This paper describes the name entity transliteration system which we conducted for the "NEWS2009 Machine Transliteration Shared Task" (Li et al 2009). We get the transliteration in Chinese from an English name with three steps. We syllabify the English name into a sequence of syllables by some rules, and generate the most probable Pinyin sequence with the mapping model of English syllables to Pinyin (EP model), then we convert the Pinyin sequence into a Chinese character sequence with the mapping model of Pinyin to characters (PC model). And we get the final Chinese character sequence. Our system achieves an ACC of 0.498 and a Mean F-score of 0.786 in the official evaluation result.

1 Introduction

The main subject of shared task is to translate English names (source language) to Chinese names (target language). Firstly, we fix some rules and syllabify the English names into a sequence of syllables by these rules, in the meanwhile, we convert the Chinese names into Pinyin sequence. Secondly, we construct an EP model referring to the method of phrase-based machine translation. In the next, we construct a 2-gram language model on characters and a chart reflecting the using frequency of each character with the same pronunciation, both of which constitute the PC model converting Pinyin sequence into character sequence. When a Pinyin is mapped to several different characters, we can use them to make a choice. In our experiment, we adopt the corpus provided by NEWS2009 (Li et al 2004) and the LDC Name Entity Lists [1] respectively to conduct two EP models, while the NEWS2009 corpus for the PC model. The experiment indicates that the larger a training corpus is, the more precise the transliteration is.

2 Transliteration System Description

Knowing from the definition of transliteration, we must make the translating result maintain the original pronunciation in source language. We found that most English letters and letter compositions' pronunciation are relatively fixed, so we can take a syllabification on an English name, therefore the syllable sequence can represent its pronunciation. In Chinese, Pinyin is used to represent a character's pronunciation. Based on these analyses, we transliterate the English syllable sequence into a Pinyin sequence, and then translate the Pinyin sequence into characters.

We suppose that the probability of a transliteration from an English name to a Chinese name is denoted by P(Ch|En), the probability of a translation from an English syllable sequence to a Pinyin sequence is denoted by P(Py|En), and the probability of a translation from a Pinyin sequence to a characters is denoted by P(Ch|Py), then we can get the formula:

$$P(Ch|En) = P(Ch|Py) * P(Py|En) \quad (1)$$

The character sequence in candidates having the max value of P(Ch|En) is the best transliteration(Wan and Verspoor, 1998).

2.1 Syllabification of English Names

English letters can be divided into vowel letters (VL) and consonant letters (CL). Usually, in a

[1]: Chinese <-> English Name Entity Lists v 1.0, LDC Catalog No.: LDC2005T34

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 96–99,
Suntec, Singapore, 7 August 2009.

word, a phonetic syllable can be constructed in a structure of CL+VL, CL+VL+CL, CL+VL+NL. To adapt for Chinese phonetic rule, we divide the continuous CLs into independent CLs(IC) and divide structure of CL+VL+CL into CL+VL and an IC. Take "Ronald" as an example, it can be syllabified into "Ro/na/l/d", "Ro" is CL+VL, "nal" is CL+VL+CL, and is divided into CL+VL and IC. 'd' is an independent CL(KUO et al. 2007). Of course there are some English names more complex to be syllabified, so we define seven rules for syllabification (JIANG et al. 2006):

(1) Define English letter set as O, vowel set as V={a, e, i, o, u}, consonant set as C=O-V.

(2) Replace all "x" in a name with "ks" before syllabification because it's always pronounced as "ks".

(3) The continuous VLs should be regarded as one VL.

(4) There are some special cases in rule (3), the continuous VLs like "oi", "io", "eo" are pronounced as two syllables, so they should be cut into two parts, so "Wilhoit" will be syllabifyd into "wi/l/ho/i/t".

(5) The continuous CLs should be cut into several independent CLs. If the last one is followed by some VLs, they will make up a syllable.

(6) Some continuous CLs are pronounced as a syllable, such as "ck", "th", these CLs will not be syllabifyd and be regarded as a single CL, "Jack" is syllabifyd into "Ja/ck".

(7) There are some other composition with the structure of VL+CL, such as "ing", "er", "an" and so on. If it's a consonant behind these compositions in the name, we can syllabify it at the end of the composition, while if it's a vowel behind them, we should double write the last letter and syllabify the word between the two same letters.

After syllabicating English names, we convert corresponding Chinese names into Pinyin. There are a few characters with multiple pronunciations in the training data, we find them out and ensure its pronunciation in a name manually.

We record all of these syllables got from the training data set, if we meet a syllable out of vocabulary when transliterating an English name, we will find a similar one with the shortest edit-distance in the vocabulary to replace that.

2.2 Mapping Model of English Syllables to Pinyins

The EP model consists of a phrase-based machine translation model with a trigram language model.

Given an English name **f**, we want to find its Chinese translation **e**, which maximize the conditional probability $\Pr(e \mid f)$, as shown below.

$$e^* = \arg\max_e \Pr(e \mid f) \quad (2)$$

Using Bayes rule, (1) can be decomposed into a Translation Model $\Pr(f \mid e)$ and a Language Model $\Pr(e)$ (Brown et al. 1993), which can both be trained separately. These models are usually regarded as features and combined with scaling factors to form a log-linear model (Och and Ney 2002). It can then be written as:

$$\Pr(e \mid f) = p_{\lambda_1^M}(e \mid f) = \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(e, f)]}{\sum_{e'} \exp[\sum_{m=1}^{M} \lambda_m h_m(e', f)]} \quad (3)$$

In our model, we use the following features:

- phrase translation probability $p(\bar{e} \mid \bar{f})$
- lexical weighting $lex(\bar{e} \mid \bar{f})$
- inverse phrase translation probability $p(\bar{f} \mid \bar{e})$
- inverse lexical weighting $lex(\bar{f} \mid \bar{e})$
- phrase penalty (always exp(1) = 2.718)
- word penalty (target name length)
- target language model, trigram

The first five features can be seen as a whole phrase translation cost and used as one during decoding.

In general, the translation process can be described as follows:

(1). Segmenting input English syllable sequence **f** into J syllables $\bar{f}_1^J$

(2). Translating each English syllable $\bar{f}_j$ into several Pinyins $\bar{e}_{jk}$

(3). Selecting the N-best words $e_1 ... e_n$, combined with reordering and Language Model and other features

(4). Rescoring the translation word set with additional features to find the best one.

We use SRI toolkit to train our trigram language model with modified Kneser-Ney smoothing (Chen and Goodman 1998). In the standard experiment, we use training data set provided by NEWS2009 (Li et al 2004) to train this language model, in the nonstandard one, we use that and the LDC Name Entity Lists to train this language model.

2.3 Mapping Model of Pinyins to Chinese Characters

Since the Chinese characters used in people names are limited, most of the conversions from Pinyin to character are fixed. But some Pinyins still have several corresponding characters, and we should make a choice among these characters. To solve this problem, we conduct a PC model consisting a frequency chart which reflects the using frequency of each character at different positions in the names and a 2-gram language model with absolute discounting smoothing.

A Chinese name is represented as $C_1C_2 \cdots C_n$, C_i ($1 \leqslant i \leqslant n$) is a Chinese character. C_1 is at the first position, we call it FW; $C_2 \cdots C_{n-1}$ are in the middle, we call them MW; C_n is at the last position, we call it LW. Usually, each character has different frequencies at these three positions. In the training data set of NEWS2009, Pinyin "luo" can be mapped to three characters: "罗", "洛", and "萝", each of them has different frequencies at different positions.

	FW	MW	LW
罗	0.677	0.647	0.501
洛	0.323	0.352	0.499
萝	0	0.001	0

Table 1. Different frequencies at different positions

From this table, we can see that at FW and MW position, "罗" is more probable to be chosen than the others, but sometimes "洛" or "萝" is the correct one. In order to ensure characters with lower frequency like "洛" and "萝" can be chosen firstly in a certain context, we conduct a 2-gram language model.

If a Pinyin can be mapped to several characters, the condition probability ($P(Ch_i|py)$) indicating that how possible a character should be chosen is determined by the weighted average of its position frequency ($P(Ch_i|pos)$) and its probability in the 2-gram language model ($P(Ch_i|Ch_{i-1})$).

$$P(Ch_i|py) = a*P(Ch_i|pos)+(1-a)*P(Ch_i|Ch_{i-1}) \quad (4)$$

$0 < a < 1$. In our experiments, we set $a = 0.1$.

2.4 Experiments and Results

We carried out two experiments. The difference between them is the training data for EP model. The standard experiment adopts corpus provided by NEWS2009, while the nonstandard one adopts LDC Name Entity Lists.

Corpora	Name Num
LDC2005T34	572213
NEWS09_train_ench_31961	31961

Table 2. Corpora used for training the EP model

Considering that an English name may be translated to different Chinese names in different corpora, so we established a unique PC model with the training data set provided by NEWS2009 to avoid the model's deviation caused by different corpora.

The experimenting data is the development data set provided by NEWS2009 (Li et al 2004), testing script is also provided by NEWS2009.

First, we take a syllabification on testing names. Then we use the EP model to generate 5-best Pinyin sequences and their probabilities. For each Pinyin sequence, the PC model gives 3-best character sequences and their probabilities. In the end, we sort the results by probabilities of character sequences and corresponding Pinyin sequences.

The evaluation results are shown below.

Metrics	Standard	Nonstandard
ACC	0.490677	0.502417
Mean F-score	0.782039	0.784203
MRR	0.606424	0.611214
MAP_ref	0.490677	0.502417
MAP_10	0.189290	0.189782
MAP_sys	0.191476	0.192129

Table 3. Evaluation results of standard and nonstandard experiments

It's easy to see that nonstandard test is better than standard one on each metric. A larger corpus does make a contribution to a more accurate model.

For the official evaluation, we make two tests on the testing data set provided by NEWS2009 (Li et al 2004). The table 4 shows respectively the evaluation results of standard and nonstandard tests given by NEWS2009.

Metrics	Standard	Nonstandard
ACC	0.498	0.500
Mean F-score	0.786	0.786
MRR	0.603	0.607
MAP_ref	0.498	0.500
MAP_10	0.187	0.189
MAP_sys	0.189	0.191

Table 4. Official evaluation results of standard and nonstandard tests

3 Conclusion

We construct a name entity transliteration system based on syllable. This system syllabifies English names by rules, then translates the syllables to Pinyin and Chinese characters by statistics model. We found that a larger corpus may improve the transliteration. Besides, we can do something else to improve that. We need to fix more complex rules for syllabification. If we can get the name user's gender from some features of the name itself, then translate the male and female names on different Chinese character sets, the results may be more precise.

Acknowledgments

This work was supported by the National Science Foundation of China (60736044, 60773027), as well as 863 Hi-Tech Research and Development Program of China (2006AA010108

-5, 2008AA01Z145).

We also thank Haizhou Li, Min Zhang and Jian Su for providing the English-Chinese data.

Reference

Franz Josef Och and Hermann Ney. 2002. "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation". In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Haizhou Li, A Kumaran, Min Zhang, Vladimir Pervouchine, "Whitepaper of NEWS 2009 Machine Transliteration Shared Task". In *Proceedings of ACL-IJCNLP 2009 Named Entities Workshop (NEWS 2009)*, Singapore, 2009

Haizhou Li, Min Zhang, Jian Su. 2004. "A joint source channel model for machine transliteration", In *Proceedings of the 42nd ACL*, 2004

Jiang Long, Zhou Ming, and Chien Lee-feng. 2006. "Named Entity Translation with Web Mining and Transliteration". *Journal of Chinese Information Processing*, 21(1):1629--1634.

Jin-Shea Kuo, Haizhou Li, and Ying-Kuei Yang. 2007. "A Phonetic Similarity Model for Automatic Extraction of Transliteration Pairs". *ACM Trans. Asian Language Information Processing*, 6(2), September 2007.

Peter F. Brown, Stephen A. Della Pietra, et al. 1993. "The Mathematics of Statistical Machine Translation: Parameter Estimation". *Computational Linguistics* 19(2): 263-311.

Stanley F. Chen and Joshua Goodman. 1998. "An empirical study of smoothing techniques for language modeling". *Technical Report* TR-10-98, Harvard University.

Stephen Wan and Cornelia Maria Verspoor. 1998. "Automatic English-Chinese name transliteration for development of multilingual resources". In *Proceedings of the 17th international conference on Computational linguistics*, 2: 1352 – 1356.