English-Korean Named Entity Transliteration Using Statistical Substring-based and Rule-based Approaches

Yu-Chun Wang
Department of Computer Science
and Information Engineering
National Taiwan University, Taiwan
d97023@csie.ntu.edu.tw

Richard Tzong-Han Tsai
Department of Computer Science
and Engineering
Yuan Ze University, Taiwan
thtsai@saturn.yzu.edu.tw

Abstract

This paper describes our approach to English-Korean transliteration in NEWS 2011 Shared Task on Machine Transliteration. We adopt the substring-based transliteration approach which group the characters of named entity in both source and target languages into substrings and then formulate the transliteration as a sequential tagging problem to tag the substrings in the source language with the substrings in the target language. The CRF algorithm are used to deal with this tagging problem. We also construct a rule-based transliteration method for comparison. Our standard and non-standard runs achieves 0.43 and 0.332 in top-1 accuracy which were ranked as the best for the English-Korean pair.

1 Introduction

Named entity translation plays an important role in machine translation, cross-language information retrieval, and question answering. However, named entities such as person names or organization names are generated everyday and do not often appear in dictionaries since bilingual dictionaries cannot update their contents frequently. Most name entity translation is based on transliteration, which is a method to map phonemes or graphemes from source language into target language. Therefore, it is necessary to construct a named entity transliteration system.

For English-Korean name entity transliteration, we adopt the substring-based transliteration proposed by Reddy and Waxmonsky (Reddy and Waxmonsky, 2009) with conditional random fields (CRF). The method treats the transliteration as a sequential labeling task where substring tokens in the source languages are tagged with the substring tokens in the target language with CRF. Since Korean writing system, Hangul, is alphabetic, we consider that the sequential labeling method is suitable for English-Korean transliteration. In addition, we also apply rule-based method with a pronouncing dictionary for comparison.

2 Our Approach

We comprises three different approaches for the transliteration: *grapheme substring-based*, *phoneme substring-based*, and *rule-based* methods. Grapheme and phoneme substring-based methods are both based on substring-based transliteration methods with CRF. The difference is that the substrings composed with English characters or English phonemes. The details of each methods are described in the following subsections.

2.1 Substring-based Approach

The substring-based approach comprise the following steps:

1. Pre-processing
2. Substring alignment
3. CRF training
4. Substring segmentation and transliteration

2.1.1 Pre-processing

Korean writing system, namely *Hangul*, is alphabetical. However, unlike western writing system with Latin alphabets, Korean alphabet is composed into syllabic blocks. For transliteration from other languages to Korean, one syllabic block contains two or three letters mainly, including 14 leading consonants, 10 vowels, and 7 tailing consonants. For instance, the syllabic block "한" (han) is composed with three letters: a leading consonant "ㅎ" (h), a vowel "ㅏ" (a), and a tailing consonant "ㄴ" (n).

Proceedings of the 2011 Named Entities Workshop, IJCNLP 2011, pages 32–35,
Chiang Mai, Thailand, November 12, 2011.

Thus, in order to deal with Korean training data, we have to decompose Korean syllabic blocks into letters before performing training. The Korean letters in syllabic blocks are almost perfectly corresponding to their phonological forms. However, the actual pronunciation of some consonant letters may vary in different positions in the syllabic block. For example, the letter "ㅅ" is pronounced as [s] in the leading consonant position, but as [t] in the tailing consonant position. We do not distinguish this pronunciation difference of these letters and treat them as the same tokens. For convenient processing, we convert the Korean letters into Roman symbols with the Revised Romanization of Korean proposed by the South Korea Government.

2.1.2 Substring alignment

Unlike Korean, English orthography might not reflect its actual phonological forms, which makes trivial one-to-one character alignment between English and Korean not practical. English may use several characters for one phoneme which is presented in one letter in Korean, such as "ch" to "ㅊ" and "oo" to "ㅜ". In contrast, English sometimes use a single character for a diphthong or consonant cluster, which are presented as several letters in Korean. For example, the letter "x' in the English name entity "Texas" corresponds to two letters "ㄱ" and "ㅅ" in Korean. Besides, some English letters in the word might not be pronounced, like "k" in the English word "knight".

Furthermore, due to Korean phonology, Korean may insert a specific vowel "ㅡ" [ɯ] between English consonant clusters or behind the last burst stop consonant of the syllable. For instance, the English name entity "Snell" is transliterated as "스넬" /s**ɯ** nel/ and "Albert" is transliterated as "앨버트" /æl bə t^{h}**ɯ**/.

In order to deal with these complex orthography problems, we adopt substring-based method to group characters into substrings. English words are segmented into several substrings and each substring maps to a substring in the target language, Korean.

To create training sets of substrings, we use the GIZA++ toolkit (Och and Ney, 2003) to align all the name entity pairs in the training data. The GIZA++ toolkit performs one-to-many alignments, which means that a single symbol in the source language may be aligned to at least one symbol in the target language. To obtain the many-to-many substring alignments, we run GIZA++ on the data in both directions from source language to target language and target language to source language. The final bidirectional alignment result is the union of the alignments in both directions. Inserted characters (aligned to NULL by GIZA++) in the alignment results are merged with the preceding character into the same substring. For example, the bidirectional alignment result of the English word "*KNOX*" to the Korean word "*nok sɯ*" (녹스) is [KN → n, O → o, X → k, *null* → s, *null* → ɯ]. The *null* → s and *null* → ɯ mappings are merged into the previous alignment to generate X → ksɯ. Finally, we get the one-to-one alignment as [KN → n, O → o, X → ksɯ].

After the processing of the bidirectional alignments, we transform the training data into one-to-one substring mapping pairs. These substrings pairs are used as token set fro the CRF training. A few pairs in the training data cannot be aligned one-to-one such as "THAILAND" to /t^{h}a i/ (타이) because they are not actual transliterations. We drop these pairs from the training data because CRF can handle one-to-one alignments only.

In addition, since Korean is a phonological writing system, for non-standard runs, we also adopt phonemic information for English name entities. The English word pronunciations are obtained from the CMU Pronouncing Dictionary v0.7a[1]. The CMU pronouncing dictionary provides the phonemic representations of English pronunciations with a sequence of phoneme symbols. For instance, the English word *KNOX* is segmented and tagged as the phonemic representation < N AA K S >. Since the CMU pronouncing dictionary does not cover all the pronunciation information of the name entities in the training data, we also apply LOGIOS Lexicon Tool[2] to generate the phonemic representations of all other name entities not in the CMU pronouncing dictionary. After obtaining the phonemic representation of all the English named entities in the training data, we formulate the sequence of phoneme symbols of the English name entities as a string and apply the substring alignment method mentioned earlier to get the mappings from English phoneme symbols to Korean letters. For the previous example, the phoneme symbols < N AA K S > from the English name entity *KNOX* are aligned to the letters

[1] http://www.speech.cs.cmu.edu./cgi-bin/cmudict

[2] http://www.speech.cs.cmu.edu/tools/lextool.html

of its corresponding Korean word "*nok sɯ*" as [N → n, AA → o, K → k, S → sɯ]. We name this substring alignment based on the English phonemic representation as "phoneme substring-based" method for non-standard run, and the substring alignment based on the English orthography as "grapheme substring-based" for standard run.

2.1.3 CRF training

With the transformed substring training data, we now use CRF to train a sequential model with the substrings as the basic tokens. We adopt the CRF++ open-source toolkit (Kudo, 2005).

We train our CRF models with the unigram , bigram, and trigram features over the input substrings in the source language. The features are shown in the following.

- Unigram: s_{-1}, s_0, and s_1
- Bigram: $s_{-1}s_0$
- Trigram: $s_{-2}s_{-1}s_0$, $s_{-1}s_0s_1$, and $s_0s_1s_2$

where current substring is s_0 and s_i is other substrings relative to the position of the current substring.

2.1.4 Substring segmentation and transliteration

Because our method is based on the substrings from the transformed training data, we have to segment the unseen English named entities into the substrings before applying CRF testing of our model. For example, we have to segment the English named entity "SHASHI" into four substrings $<$ SH A SH I $>$. Since the substrings used to train the CRF model are generated by the bidirectional alignments from the training data, we also used CRF to train another model for substring segmentation of English named entities.

We adopt the segmentation approach motivated by the Chinese segmentation (Tsai et al., 2006) which treat Chinese segmentation as a tagging problem. The characters in a sentence are tagged in **B** class if it is the first character of a Chinese word or in **I** class if it is in a Chinese word but not the first character. Thus, we collect all the substring results from the bidirectional alignments and tag each character in the English named entity in the training data as **B** class (the first character of the substring) or **I** class (not the first character of the substring) to create a training data of substring segmentation for CRF. Since each character should belong to one substring, we need only **B** and **I** classes in the tag sets.

After the English named entities are segmented into substrings, it can be passed into the CRF model we trained in section 2.1.3 as input data to produce the transliteration results.

The transliteration results predicted by the CRF model is an romanized representation of Korean letters. Therefore, the romanized representation sequences should be converted back to Korean syllabic blocks. Because the position information of each Korean letters in the syllabic blocks (leading consonant, vowel and tailing consonant mentioned in section 2.1.1) does not remain while training, we have to organize the sequential letters into blocks based on the Korean orthography. Korean orthographic rules are applied to combine the letters into syllabic blocks. For example, the sequential Korean letters "ㅁ, ㅏ, ㄱ, ㅅ, ㅣ" (m, a, k, s, i) are combined into two syllabic blocks "막시" (mak-si) to make "k" in the tailing consonant position of the first syllable and "s" in the leading consonant position of the second syllable because consonant clusters are not allowed in a Korean syllabic block. Besides, between the successive vowel letters, the zero consonant letter " ㅇ " is inserted because of Korean orthography.

2.2 Rule-based Approach

We also construct a rule-based transliteration system. According to the "외래어 표기법" (Korean writing method of loanwords)[3] standardized by the National Institute of Korean Language, we build a transliteration mapping table from international phonetic alphabet (IPA) to Korean letters. The phonemic representations of English name entities in the test set are first extracted by the CMU Pronouncing Dictionary and LOGIOS Lexicon Tool. Then, each phoneme symbol is transliterated into corresponding Korean letter based on the transliteration mapping table. The results generated by the mapping table need to be composed into Korean syllabic blocks. We use the same technique described in section 2.1.4 to produce the final results of the rule-based method.

3 Results

Table 1 shows the final results of our transliteration approaches on the test data. We construct four

[3] http://www.korean.go.kr/09_new/dic/rule/rule_foreign_0101.jsp

Run	Accuracy	Mean F-score	MRR	MAP_{ref}
Grapheme substring-based	**0.430**	**0.711**	0.430	**0.423**
Phoneme substring-based	0.332	0.653	0.332	0.325
Rule-based	0.215	0.474	0.215	0.209
Mixed	0.332	0.653	**0.467**	0.332

Table 1: Final results on the test data

runs as following.

- **Grapheme substring-based**: CRF model with the substring training set based on English orthography.
- **Phoneme substring-based**: CRF model with the substring training set based on English phonemic representations.
- **Rule-based**: transliteration mapping table from English phonemes to Korean letters.
- **Mixed**: union of the results from the previous three runs in the order of Phoneme substring-based, Grapheme substring-based and Rule-based.

The results show that the grapheme-based approach achieves better than others in the four evaluation metrics. The rule-based one does not perform well due to the rules from the Korean writing method of loanwords may not be enough to cover most possible cases of the transliteration detailedly. However, the result of the phoneme substring-based approach is not as good as the grapheme substring-based one. It might be due to two reasons: one is that the Korean transliteration sometimes is based on the orthography not the actual pronunciation; the second reason is that the pronunciation from LOGIOS lexicon tool may not be accurate to get the correct phonemic forms. The phoneme substring-based and rule-based approaches suffer such problems. The performance of the mixed run which merged the results of above three runs shows that the joint result of different methods can help cover more possible transliterations.

4 Conclusion

In this paper, we adopt the substring-based transliteration approach with CRF model for English-Korean named entity transliteration. The characters in the source and target language are aligned in bi-direction and then group into substrings to generate the substring mappings from the source language to the target language. Then, the transliteration is formulated as a sequential tagging problem to tag the substrings in the source language with the substrings in the target language. The CRF algorithm is used to deal with this tagging problem. For English substring generation, we create two types of substrings. One is based on the English orthography, and the other is based on the phonemic symbols from the CMU pronouncing dictionary. In addition, we also construct a rule-based transliteration system based on the Korean writing method of loanwords from the National Institute of Korean language. From the evaluation results, the substring-based method based on the English orthography performs better than other runs.

For future work, we plan to add more phonetic features for the CRF training and try to integrate the CRF-based statistical based method and the rule-based methods to improve the transliteration performance. We also try to apply the re-ranking techniques from the web data to get better transliteration results.

References

Taku Kudo. 2005. CRF++: Yet another CRF toolkit. Available at `http://chasen.org/ttaku/software/ctf++/`.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Liguistics*, 29(4):417–449.

Sravana Reddy and Sonjia Waxmonsky. 2009. Substring-based transliteration with conditional random fields. *Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP*, pages 92–95.

Richard Tzong-Han Tsai, Hsieh-Chuan Hung, Cheng-Lung Sung, Hong-Jie Dai, , and Wen-Lian Hsu. 2006. On closed task of chinese word segmentation: An improved crf model coupled with character clustering and automatically generated template matching. *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processin*, pages 134–137.