Transliteration Mining with Phonetic Conflation and Iterative Training

Kareem Darwish
Cairo Microsoft Innovation Center
Cairo, Egypt
kareemd@microsoft.com

Abstract

This paper presents transliteration mining on the ACL 2010 NEWS workshop shared transliteration mining task data. Transliteration mining was done using a generative transliteration model applied on the source language and whose output was constrained on the words in the target language. A total of 30 runs were performed on 5 language pairs, with 6 runs for each language pair. In the presence of limited resources, the runs explored the use of phonetic conflation and iterative training of the transliteration model to improve recall. Using letter conflation improved recall by as much as 48%, with improvements in recall dwarfing drops in precision. Using iterative training improved recall, but often at the cost of significant drops in precision. The best runs typically used both letter conflation and iterative learning.

1 Introduction

Transliteration Mining (TM) is the process of finding transliterated word pairs in parallel or comparable corpora. TM has many potential applications such as building training data for training transliterators and improving lexical coverage for machine translation and cross language search via translation resource expansion. TM has been gaining some attention of late with a shared task in the ACL 2010 NEWS workshop[1]. In this paper, TM was performed using a transliterator that was used to generate possible transliterations of a word while constraining the output to tokens that exist in a target language word sequence. The paper presents the use of phonetic letter conflation and iterative transliterator training to improve TM when only limited transliteration training data is available. For phonetic letter conflation, a variant of SOUNDEX (Russell, 1918) was used to improve the coverage of existing training data. As for iterative transliterator training, an initial transliterator, which was trained on initial set of transliteration pairs, was used to mine transliterations in parallel text. Then, the automatically found transliterations pairs were considered correct and were used to retrain the transliterator.

[1] http://translit.i2r.a-star.edu.sg/news2010/

The proposed improvements in TM were tested using the ACL 2010 NEWS workshop data for Arabic, English-Chinese, English-Hindi, English-Russian, and English-Tamil. For language pair, a base set of 1,000 transliteration pairs were available for training.

The rest of the paper is organized as follows: Section 2 surveys prior work on transliteration mining; Section 3 describes the TM approach and the proposed improvements; Section 4 describes the experimental setup including the evaluation sets; Section 5 reports on experimental results; and Section 6 concludes the paper.

2 Background

Much work has been done on TM for different language pairs such as English-Chinese (Kuo et al., 2006; Kuo et al., 2007; Kuo et al., 2008; Jin et al. 2008;), English-Tamil (Saravanan and Kumaran, 2008; Udupa and Khapra, 2010), English-Korean (Oh and Isahara, 2006; Oh and Choi, 2006), English-Japanese (Brill et al., 2001; Oh and Isahara, 2006), English-Hindi (Fei et al., 2003; Mahesh and Sinha, 2009), and English-Russian (Klementiev and Roth, 2006). The most common approach for determining letter sequence mapping between two languages is using automatic letter alignment of a training set of transliteration pairs. Automatic alignment can be performed using different algorithms such as the EM algorithm (Kuo et al., 2008; Lee and Chang, 2003) or using an HMM aligner (Udupa et al., 2009a; Udupa et al., 2009b). Another method is to use automatic speech recognition confusion tables to extract phonetically equivalent character sequences to discover monolingual and cross lingual pronunciation variations (Kuo and Yang, 2005). Alternatively, letters can be mapped into a common character set. One example of that is to use a predefined transliteration scheme to transliterate a word in one character set into another character set (Oh and Choi, 2006). Different methods were proposed to ascertain if two words can be transliterations of each other. One such way is to use a generative model that attempts to generate possible transliterations given the character mappings between two character sets (Fei et al., 2003; Lee and Chang, 2003, Udupa et al., 2009a). A similar alternative is to use back-transliteration to determine if one sequence could have been generated by successively mapping character sequences from one language into another (Brill et al., 2001; Bilac and Tanaka, 2005; Oh and Isahara, 2006). Another mapping method is to map candidate transliterations into a common representation space (Udupa et al., 2010). When using a predefined transliteration scheme, edit distance was used to determine if candidate transliterations were indeed transliterations (Oh and Choi, 2006). Also letter conflation was used to find transliterations (Mahesh and Sinha, 2009). Different methods were proposed to improve the recall of mining. For example, Oh and Choi (2006) used a SOUNDEX like scheme to minimize the effect of vowels and different schemes of phonetically coding names. SOUNDEX is used to convert English words into a simplified phonetic representation, in which vowels are removed and phonetically similar characters are conflated. Another method involved expanding character sequence maps by automatically mining transliteration pairs and then aligning these pairs to generate an expanded set of character sequence maps (Fei et al., 2003).

3 Transliteration Mining

TM proposed in this paper uses a generative transliteration model, which is trained on a set of transliteration pairs. The training involved automatically aligning character sequences. SOUNDEX like letter conflation and iterative transliterator training was used to improve recall. Akin to phrasal alignment in machine translation, character sequence alignment was treated as a word alignment problem between parallel sentences, where transliterations were treated as if they were sentences and the characters from which they were composed were treated as if they were words. The alignment was performed using a Bayesian learner that trained on word dependent transition models for HMM based word alignment (He, 2007). Alignment produced a mapping of source character sequence to a target character sequence along with the probability of source given target.

For all the work reported herein, given an English-foreign language transliteration candidate pair, English was treated as the target language and the foreign language as the source. Given a foreign source language word sequence F_1^n and an English target word sequence E_1^m, $F_i \in F_1^n$ is a potential transliteration of $E_j \in E_1^m$. Given F_i, composed of the character sequence $f_1 \ldots f_o$, and E_j, composed of the character sequence $e_1 \ldots e_p$, $P(F_i|E_j)$ is calculated using the trained model, as follows:

$$P(E_i|F_j) = \prod_{alt\ f_x \ldots f_y} P(f_x \ldots f_y | e'_k \ldots e'_l)$$

The non-overlapping segments $f_x \ldots f_y$ are generated by finding all possible 2^{n-1} segmentations of the word F_i. For example, given "man" then all possible segmentations are (m,a,n), (ma,n), (m,an), and (man). The segmentation producing the highest probability is chosen. All segment sequences $e'_k \ldots e'_l$ known to produce $f_x \ldots f_y$ for each of the possible segmentations are produced. If a set of non-overlapping sequences of $e'_k \ldots e'_l$ generates the sequence $e_1 \ldots e_p$ (word $E_j \in E_1^m$), then E_j is considered a transliteration of F_i. If multiple target words have $P(F_i|E_j) > 0$, then E_j that maximizes $P(F_i|E_j)$ is taken as the proper transliteration. A suffix tree containing E_1^m was used to constrain generation, improving efficiency. No smoothing was used.

To improve recall, a variant of SOUNDEX was used on the English targets. The original SOUNDEX scheme applies the following rules:

1. Retain the first letter in a word
2. Remove all vowels, H, and W
3. Perform the following mappings:

B, F, P, V ➔ 1	C, G, J, K, Q, S, X, Z ➔ 2
D,T ➔ 3	L ➔ 4
M,N ➔ 5	R ➔ 6

4. Trim all result sequences to 4 characters
5. Pad short sequences with zeros to have exactly 4 characters.

SOUNDEX was modified as follows:

1. The first letter in a word was not retained and was changed according the mapping in step 3 of SOUNDEX.
2. Resultant sequences longer than 4 characters were not trimmed.
3. Short resultant strings were not padded with zeros.

SOUNDEX after the aforementioned modifications is referred at S-mod. Alignment was performed between transliteration pairs where English words were replaced with their S-mod representation. Case folding was always applied to English.

Iterative transliterator training involved training a transliterator using an initial seed of transliteration pairs, which was used to automatically mine transliterations from a large set of parallel words sequences. Automatically mined transliteration pairs were assumed to be correct and were used to retrain the transliterator. S-mod and iterative training were used in isolation or in combination as is shown in the next section.

Russian and Arabic were preprocessed as follows:

- Russian: characters were case-folded
- Arabic: the different forms of *alef* (*alef*, *alef maad*, *alef with hamza on top*, and *alef with hamza below it*) were normalized to *alef*, *ya* and *alef maqsoura* were normalized to *ya*, and *ta marbouta* was mapped to *ha*.

No preprocessing was performed for the other languages. Since Wikipedia English entries often had non-English characters, the following letter conflations were performed:

ž, ż ➔ z	á, â, ä, à, ã, ā, ą, æ ➔ a
é, ę, è ➔ e	ć, č, ç ➔ c
ł ➔ l	ï, í, ì, î ➔ i
ó, ō, ö, õ ➔ o	ń, ñ, ṅ ➔ n
ş, ś, ß, š ➔ s	ř ➔ r
ý ➔ y	ū, ü, ú, û ➔ u

Language Pair	**# of Parallel Sequences**
English-Arabic	90,926
English-Chinese	196,047
English-Hindi	16,963
English-Russian	345,969
English-Tamil	13,883

Table 1: Language pairs and no. of parallel sequences

Run	Precision	Recall	F-score
1	0.900	0.796	0.845
2	**0.966**	0.587	0.730
3	0.952	0.588	0.727
4	0.886	0.817	**0.850**
5	0.895	0.678	0.771
6	0.818	**0.827**	0.822

Table 2: English-Arabic mining results

Run	Precision	Recall	F-score
1	1.000	0.024	0.047
2	1.000	0.016	0.032
3	1.000	0.016	0.032
4	1.000	0.026	0.050
5	1.000	0.022	0.044
6	1.000	0.030	0.059

Table 3: English-Chinese mining results

Run	Precision	Recall	F-score
1	0.959	0.786	0.864
2	**0.987**	0.559	0.714
3	0.984	0.569	0.721
4	0.951	0.812	**0.876**
5	0.981	0.687	0.808
6	0.953	**0.855**	0.902

Table 4: English-Hindi mining results

Run	Precision	Recall	F-score
1	0.813	0.839	**0.826**
2	**0.868**	0.748	0.804
3	0.843	0.747	0.792
4	0.716	0.868	0.785
5	0.771	0.794	0.782
6	0.673	**0.881**	0.763

Table 5: English-Russian mining results

Run	Precision	Recall	F-score
1	0.963	0.604	0.743
2	**0.976**	0.407	0.575
3	0.975	0.446	0.612
4	0.952	0.668	0.785
5	0.968	0.567	0.715
6	0.939	**0.741**	**0.828**

Table 6: English-Tamil mining results

For each foreign language (F) and English (E) pair, a set of 6 runs were performed. The first two runs involved training a transliterator using the 1,000 transliteration pairs and using it for TM as in section 3. The runs were:
Run 1: align F with S-mod(E)
Run 2: align F with E
The four other runs involved iterative training in which all automatically mined transliterations from Runs 1 and 2 were considered correct, and were used to retrain the transliterator. The runs were:
Run 3: Use Run 2 output, align F with E
Run 4: Use Run 2 output, align F with S-mod(E)
Run 5: Use Run 1 output, align F with E
Run 6: Use Run 1 output, align F with S-mod(E)

For evaluation, the system would mine transliterations and a set of 1,000 parallel sequences were chosen randomly for evaluation. The figures of merit are precision, recall, F1 measure.

4 Experimental Setup

The experiments were done on the ACL-2010 NEWS Workshop TM shared task datasets. The datasets cover 5 language pairs. For each pair, a dataset includes a list of 1,000 transliterated words to train a transliterator, and list of parallel word sequences between both languages. The parallel sequences were extracted parallel Wikipedia article titles for which cross language links exist between both languages. Table 1 lists the language pairs and the number of the parallel word sequences.

5 Experimental Results

Tables 2, 3, 4, 5, and 6 report results for Arabic, Chinese, Hindi, Russian and Tamil respectively. As shown in Table 3, the recall for English-Chinese TM was dismal and suggests problems in experimental setup. This would require further investigation. For the other 4 languages, the results show that not using S-mod and not using iterative training, as in Run 2, led to the highest precision. Using both S-mod and iterative training, as in Run 6, led to the highest recall.
In comparing Runs 1 and 2, where 1 uses S-mod and 2 does not, using S-mod led to 35.6%, 40.6%, 12.2%, and 48.4% improvement in recall and to 6.8%, 2.8%, 6.3%, and 1.3% decline in precision for Arabic, Chinese, Russian, and Tamil respectively. Except for Russian, the improvements in recall dwarf decline in precision, leading to overall improvements in F-measure for all 4 languages.
In comparing runs 2 and 3 where iterative training is used, iterative training had marginal impact on precision and recall. When using S-mod, comparing run 6 where iterative training was performed over the output from run 1, recall increased by

3.9%, 8.8%, 5.0%, and 22.7% for Arabic, Chinese, Russian, and Tamil respectively. The drop in precision was 9.1% and 17.2% for Arabic and Russian respectively and marginal for Hindi and Tamil. Except for Russian, the best runs for all languages included the use of S-mod and iterative training. The best runs were 4 for Arabic and Hindi and 6 for Tamil. For Russian, the best runs involved using S-mod only without iterative training. The drop in Russian could be attributed to the relatively large size of training data compared to the other languages (345,969 parallel word sequences).

6 Conclusion

This paper presented two methods for improving transliteration mining, namely phonetic conflation of letters and iterative training of a transliteration model. The methods were tested using on the ACL 2010 NEWS workshop shared transliteration mining task data. Phonetic conflation of letters involved using a SOUNDEX like conflation scheme for English. This led to much improved recall and general improvements in F-measure. The iterative training of the transliteration model led to improved recall, but recall improvements were often offset by decreases in precision. However, the best experimental setups typically involved the use of both improvements.

The success of phonetic conflation for English may indicate that similar success may be attained if phonetic conflation is applied to other languages. Further, the use of smoothing of the transliteration model may help improve recall. The recall for transliteration mining between English and Chinese were dismal and require further investigation.

References

Slaven Bilac, Hozumi Tanaka. Extracting transliteration pairs from comparable corpora. NLP-2005, 2005.

Eric Brill, Gary Kacmarcik, Chris Brockett. Automatically harvesting Katakana-English term pairs from search engine query logs. NLPRS 2001, pages 393–399, 2001.

Huang Fei, Stephan Vogel, and Alex Waibel. 2003. Extracting Named Entity Translingual Equivalence with Limited Resources. TALIP, 2(2):124–129.

Xiaodong He, 2007. Using Word-Dependent Transition Models in HMM based Word Alignment for Statistical Machine Translation. ACL-07 2nd SMT workshop.

Chengguo Jin, Dong-Il Kim, Seung-Hoon Na, Jong-Hyeok Lee. 2008. Automatic Extraction of English-Chinese Transliteration Pairs using Dynamic Window and Tokenizer. Sixth SIGHAN Workshop on Chinese Language Processing, 2008.

Alexandre Klementiev and Dan Roth. 2006. Named Entity Transliteration and Discovery from Multilingual Comparable Corpora. HLT Conf. of the North American Chapter of the ACL, pages 82–88.

Jin-Shea Kuo, Haizhou Li, Ying-Kuei Yang. 2006. Learning Transliteration Lexicons from the Web. COLING-ACL2006, Sydney, Australia, 1129 – 1136.

Jin-shea Kuo, Haizhou Li, Ying-kuei Yang. A phonetic similarity model for automatic extraction of transliteration pairs. TALIP, 2007

Jin-Shea Kuo, Haizhou Li, Chih-Lung Lin. 2008. Mining Transliterations from Web Query Results: An Incremental Approach. Sixth SIGHAN Workshop on Chinese Language Processing, 2008.

Jin-shea Kuo, Ying-kuei Yang. 2005. Incorporating Pronunciation Variation into Extraction of Transliterated-term Pairs from Web Corpora. Journal of Chinese Language and Computing, 15 (1): (33-44).

Chun-Jen Lee, Jason S. Chang. Acquisition of English-Chinese transliterated word pairs from parallel-aligned texts using a statistical machine transliteration model. Workshop on Building and Using Parallel Texts, HLT-NAACL-2003, 2003.

R. Mahesh, K. Sinha. 2009. Automated Mining Of Names Using Parallel Hindi-English Corpus. 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009, pages 48–54, Suntec, Singapore, 2009.

Jong-Hoon Oh, Key-Sun Choi. 2006. Recognizing transliteration equivalents for enriching domain-specific thesauri. 3rd Intl. WordNet Conf. (GWC-06), pages 231–237, 2006.

Jong-Hoon Oh, Hitoshi Isahara. 2006. Mining the Web for Transliteration Lexicons: Joint-Validation Approach. pp.254-261, 2006 IEEE/WIC/ACM Intl. Conf. on Web Intelligence (WI'06), 2006.

Raghavendra Udupa, K. Saravanan, Anton Bakalov, and Abhijit Bhole. 2009a. "They Are Out There, If You Know Where to Look": Mining Transliterations of OOV Query Terms for Cross-Language Information Retrieval. ECIR-2009, Toulouse, France, 2009.

Raghavendra Udupa, K. Saravanan, A. Kumaran, and Jagadeesh Jagarlamudi. 2009b. MINT: A Method for Effective and Scalable Mining of Named Entity Transliterations from Large Comparable Corpora. EACL 2009.

Raghavendra Udupa and Mitesh Khapra. 2010. Transliteration Equivalence using Canonical Correlation Analysis. ECIR-2010, 2010.

Robert Russell. 1918. Specifications of Letters. US patent number 1,261,167.

K Saravanan, A Kumaran. 2008. Some Experiments in Mining Named Entity Transliteration Pairs from Comparable Corpora. The 2nd Intl. Workshop on Cross Lingual Information Access addressing the need of multilingual societies, 2008.