Transliteration of Name Entity via Improved Statistical Translation on Character Sequences

Yan Song   Chunyu Kit   Xiao Chen
Department of Chinese, Translation and Linguistics
City University of Hong Kong
83 Tat Chee Ave., Kowloon, Hong Kong
Email: {yansong, ctckit}@cityu.edu.hk, cxiao2@student.cityu.edu.hk

Abstract

Transliteration of given parallel name entities can be formulated as a phrase-based statistical machine translation (SMT) process, via its routine procedure comprising training, optimization and decoding. In this paper, we present our approach to transliterating name entities using the log-linear phrase-based SMT on character sequences. Our proposed work improves the translation by using bidirectional models, plus some heuristic guidance integrated in the decoding process. Our evaluated results indicate that this approach performs well in all standard runs in the NEWS2009 Machine Transliteration Shared Task.

1 Introduction

To transliterate a foreign name into a target language, a direct instrument is to make use of existing rules for converting text to syllabus, or at least a phoneme base to support such transformation. Following this path, the well developed noisy channel model used for transliteration usually set an intermediate layer to represent the source and target names by phonemes or phonetic tags (Knight and Graehl, 1998; Virga and Khudanpur, 2003; Gao et al., 2004). Having been studied extensively though, the phonemes-based approaches cannot break its performance ceiling for two reasons (Li et al., 2004): (1) Language-dependent phoneme representation is not easy to obtain; (2) The phonemic representation to source and target names usually causes error spread.

Several approaches have been proposed for direct use of parallel texts for performance enhancement (Li et al., 2004; Li et al., 2007; Goldwasser and Roth, 2008). There is no straightforward mean for grouping characters or letters in the source or target language into better transliteration units for a better correspondence. There is no consistent deterministic mapping between two languages either, especially when they belong to different language families, such as English and Chinese. Usually, a single character in a source name is not enough to form a phonetic pattern in a target name. Thus a better way to model transliteration is to map character sequences between source and target name entities. The mapping is actually an alignment process. If a certain quantity of bilingual transliterated entities are available for training, it is a straight-forward idea to tackle this transliteration problem with a mature framework such as phrase-based SMT. It can be considered a general statistical translation task if the character sequences involved are treated like phrases.

In so doing, however, a few points need to be highlighted. Firstly, only parallel data are required for generating transliteration outputs via SMT, and this SMT translation process can be easily integrated as a component into a general-purpose SMT system. Secondly, on character sequences, the mapping between source and target name entities can be performed on even larger units. Consequently, contextual information can be exploited to facilitate the alignment, for a string can be used as a context for every one of its own characters. It is reasonable to expect such relevant information to produce more precisely statistical results for finding corresponding transliterations. Thirdly, transliteration as a monotonic word ordering transformation problem allows the alignment to be performed monotonously from the beginning to the end of a text. Thus its decoding is easy to perform as its search space shrinks this way, for re-ordering is considered not to be involved, in contrast to the general SMT process.

This paper is intended to present our work on applying phrased-based SMT technologies to tackle transliteration. The following sections will report how we have carried out our experiments

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 57–60,
Suntec, Singapore, 7 August 2009. ©2009 ACL and AFNLP

for the NEWS2009 task (Li et al., 2009) and present the experimented results.

2 Transliteration as SMT

In order to transliterate effectively via a phrase based SMT process for our transliteration task, we opt for the log-linear framework (Och and Ney, 2002), a straight-forward architecture to have several feature models integrated together as

$$P(t|s) = \frac{exp[\sum_{i=1}^{n} \lambda_i h_i(s,t)]}{\sum_t exp[\sum_{i=1}^{n} \lambda_i h_i(s,t)]} \tag{1}$$

Then the transliteration task is to find the proper source and corresponding target chunks to maximize $P(t|s)$ as

$$t = \underset{t}{\text{argmax}}\, P(t|s) \tag{2}$$

In (1), $h_i(s,t)$ is a feature model formulated as a probability functions on a pair of source and target texts in logarithmic form, and λ_i is a parameter to optimize its contribution. The two most important models in this framework are the translation model (i.e., the transliteration model in our case), and the target language model. The former is defined as

$$h_i(s,t) = \log p(s,t) \tag{3}$$

where $p(s,t)$ is $p(s|t)$ or $p(t|s)$ according to the direction of training corresponding phrases. (Och and Ney, 2002) show that $p(t|s)$ gives a result comparable to $p(s|t)$, as in the source-channel framework. (Gao et al., 2004) also confirm on transliteration that the direct model with $p(t|s)$ performs well while working on the phonemic level. For our task, we have tested these choices for $p(s,t)$ on all our development data, arriving at a similar result. However, we opt to use both $p(s|t)$ and $p(t|s)$ if they give similar transliteration quality in some language pairs. Thus we take $p(t|s)$ for our primary transliteration model for searching candidate corresponding character sequences, and $p(s|t)$ as a supplement.

In addition to the translation model feature, another feature for the language model can be described as

$$h_i(s,t) = \log p(t) \tag{4}$$

Usually the n-gram language model is used for its effectiveness and simplicity.

2.1 Training

For the purpose of modeling the training data, the characters from both the source and target name entities for training are split up for alignment, and then phrase extraction is conducted to find the mapping pairs of character sequence.

The alignment is performed by expectation-maximization (EM) iterations in the IBM model-4 SMT training using the GIZA++ toolkit[1]. In some runs, however, e.g., English to Chinese and English to Korean transliteration, the character number of the source text is always more than that of the target text, the training conducted only on characters may lead to many abnormal fertilities and then affect the character sequence alignment later. To alleviate this, a pre-processing step before GIZA++ training applies unsupervised learning to identify many frequently co-occurring characters as fixed patterns in the source texts, including all available training, development and testing data. All possible tokens of the source names are considered.

Afterwards, the extraction and probability estimation of corresponding sequences of characters or pre-processed small tokens aligned in the prior step is performed by 'diag-growth-final' (Koehn et al., 2003), with maximum length 10, which is tuned on development data, for both the source-to-target and the target-to-source character alignment. Then two transliteration models, namely $p(t|s)$ and $p(s|t)$, are generated by such extraction for each transliteration run.

Another component involved in the training is an n-gram language model. We set $n = 3$ and have it trained with the available data of the target language in question.

2.2 Optimization

Using the development sets for the NEWS2009 task, a minimum error rate training (MERT) (Och, 2003) is applied to tune the parameters for the corresponding feature models in (1). The training is performed with regard to the mean F-score, which is also called fuzziness in top-1, measuring on average how different the top transliteration candidate is from its closest reference. It is worth noting that a high mean F-score indicates a high accuracy of top candidates, thus a high mean reciprocal rank (MRR), which is used to quantify the overall performance of transliteration.

[1] http://code.google.com/p/giza-pp/

Table 1: Comparison: baseline v.s. optimized performance on EnCh and EnRu development sets.

		λ_1[a]	λ_2	λ_3	Mean F	MRR
EnCh[b]	B[c]	1	1	1	0.803	0.654
	O	2.38	0.33	0.29	0.837	0.709
EnRu	B	1	1	1	0.845	0.485
	O	2.52	0.27	0.21	0.927	0.687

[a] The subscripts 1, 2 and 3 refer to the two transliteration models $p(t|s)$ and $p(s|t)$ and another language model respectively, and normalized as $\sum_{i=1}^{3} \lambda_i = 3$.

[b] EnCh stands for English to Chinese run and EnRu for English to Russian run.

[c] B stands for baseline configuration and O for optimized case.

As shown in Table 1, the optimization of the three major models leads to a significant performance improvement, especially when training data is limited, such as the EnRu run, only 5977 entries of name entities are provided for training. And, it is also found that the optimized feature weights for other language pairs are similar to these for the two runs as shown in the table above[2].

Note for the optimization of the parameters, that only the training data is used for construction of models. For the test, both the training and the development sets are used for training.

2.3 Decoding

The trained source-to-target and target-to-source transliteration models are integrated with the language model as given in (1) for our decoding. We implement a beam-search decoder to deal with these multiple transliteration models, which takes both the forward- and backward-directional aligned character sequences as factors to contribute to the transliteration probability. Considering the monotonic transformation order, the decoding is performed sequentially from the beginning to the end of a source text. No re-ordering is needed for such transliteration. As the search space is restricted in this way, the accuracy of matching possible transliteration pairs is not affected when the decoding is maintained at a faster speed than that for ordinary translation. In addition, another heuristic condition is also used to guide this monotonic decoding. For those target character sequences found in the training data, their positions in a name entity can help the decoding to find better corresponding transliterations, for some texts appear more frequently at the beginning of a name entity and others at the end. We use the probabilities for all aligned target character sequences in different positions, and exploit the data as an auxiliary feature model for the generation. Finally, all possible target candidates are generated by (2) for source names.

Table 3: Numbers of name entities in NEWS2009 training data[6].

EnCh	34857	EnHi	10990
EnJa	29811	EnTa	9031
EnKo	5838	EnKa	9040
JnJk	19891	EnRu	6920

3 Evaluation Results

For NEWS2009, we participated in all 8 standard runs of transliteration task, namely, EnCh (Li et al., 2004), EnJa, EnKo, JnJk[3], EnHi, EnTa, EnKa and EnRu (Kumaran and Kellner, 2007). Ten best candidates generated for each source name are submitted for each run. The transliteration performance is evaluated by the official script[4], using six metrics[5]. The official evaluation results for our system are presented in Table 2.

The effectiveness of our approach is revealed by the fact that many of our Mean F-scores are above 0.8 for various tasks. These high scores suggest that our top candidates are close to the given references. Besides, it is also interesting to look into how well the desired targets are generated under a certain recall rate, by examining if the best answers are among the ten candidates produced for each source name. If the recall rate goes far beyond MRR, it can be a reliable indication that the desired targets are found for most source names, but just not put at the top of the ten-best. From the last column in Table 2, we can see a great chance to improve our performance, especially for EnCh, JnJk and EnRu runs.

[2] Interestingly, the first model contributes much more than others. It can achieve a comparable result even without model 2 and 3, according to our experiments.

[3] http://www.cjk.org

[4] https://translit.i2r.a-star.edu.sg/news2009/evaluation/

[5] The six metrics are Word Accuracy in Top-1 (ACC), Fuzziness in Top-1 (Mean F-score), Mean Reciprocal Rank (MRR), Precision in the n-best candidates (Map_ref), Precesion in the 10-best candidates (Map_10) and Precision in the system produced candidates (Map_sys).

[6] Note that in some of the runs, when a source name has multiple corresponding target names, the numbers are calculated according to the total target names in both the training and development data.

Table 2: Evaluation result of NEWS2009 task.

Task	Source	Target	ACC	Mean F	MRR	Map_ref	Map_10	Map_sys	Recall
EnCh	English	Chinese	0.643	0.854	0.745	0.643	0.228	0.229	0.917
EnJa	English	Katakana	0.406	0.800	0.529	0.393	0.180	0.180	0.786
EnKo	English	Hangul	0.332	0.648	0.425	0.331	0.134	0.135	0.609
JnJk	Japanese	Kanji	0.555	0.708	0.653	0.538	0.261	0.261	0.852
EnHi	English	Hindi	0.349	0.829	0.455	0.341	0.151	0.151	0.681
EnTa	English	Tamil	0.316	0.848	0.451	0.307	0.154	0.154	0.724
EnKa	English	Kannada	0.177	0.799	0.307	0.178	0.109	0.109	0.576
EnRu	English	Russian	0.500	0.906	0.613	0.500	0.192	0.192	0.828

But still, since SMT is a data-driven approach, the amount of training data could affect the transliteration results significantly. Table 3 shows the training data size in our task. It gives a hint on the connections between the performance, especially Mean F-score, and the data size. In spite of the low ACC, EnKa test has a Mean F-score close to other two runs, namely EnHi and EnTa, of similar data size. For EnRu test, although the training data is limited, the highest Mean F-score is achieved thanks to the nice correspondence between English and Russian characters.

4 Conclusion

In this paper we have presented our recent work to apply the phrase-based SMT technology to name entity transliteration on character sequences. For training, the alignment is carried out on characters and on those frequently co-occurring character sequences identified by unsupervised learning. The extraction of bi-directional corresponding source and target sequence pairs is then performed for the construction of our transliteration models. In decoding, a beam search decoder is applied to generate transliteration candidates using both the source-to-target and target-to-source transliteration models, the target language model and some heuristic guidance integrated. The MERT is applied to tune the optimum feature weights for these models. Finally, ten best candidates are submitted for each source name. The experimental results confirm that our approach is effective and robust in the eight runs of the NEWS2009 transliteration task.

Acknowledgments

The research described in this paper was supported by City University of Hong Kong through the Strategic Research Grants (SRG) 7002267 and 7002388.

References

W. Gao, K. F. Wong, and W. Lam. 2004. Improving transliteration with precise alignment of phoneme chunks and using context features. In *Proceedings of AIRS-2004*.

Dan Goldwasser and Dan Roth. 2008. Transliteration as constrained optimization. In *Proceedings of EMNLP-2008*, pages 353–362, Honolulu, USA, October.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Pharaoh: A beam search decoder for phrase-base statistical machine translation models. In *Proceedings of the 6th AMTA*, Edomonton, Canada.

A Kumaran and Tobias Kellner. 2007. A generic framework for machine transliteration. In *Proceedings of the 30th SIGIR*.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of ACL-04*, pages 159–166, Barcelona, Spain, July.

Haizhou Li, Khe Chai Sim, Jin-Shea Kuo, and Minghui Dong. 2007. Semantic transliteration of personal names. In *Proceedings of ACL-07*, pages 120–127, Prague, Czech Republic, June.

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report on news 2009 machine transliteration shared task. In *Proceedings of ACL-IJCNLP 2009 Named Entities Workshop*, Singapore.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL-02*, pages 295–302, Philadelphia, USA, July.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL-03*, pages 160–167, Sapporo, Japan, July.

Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, pages 57–64, Sapporo, Japan, July.