Transliteration by Bidirectional Statistical Machine Translation

Andrew Finch
NICT
2-2-2 Hikaridai
Keihanna Science City
619-0288 JAPAN
andrew.finch@nict.go.jp

Eiichiro Sumita
NICT
2-2-2 Hikaridai
Keihanna Science City
619-0288 JAPAN
eiichiro.sumita@nict.go.jp

Abstract

The system presented in this paper uses phrase-based statistical machine translation (SMT) techniques to directly transliterate between all language pairs in this shared task. The technique makes no language specific assumptions, uses no dictionaries or explicit phonetic information. The translation process transforms sequences of tokens in the source language directly into to sequences of tokens in the target. All language pairs were transliterated by applying this technique in a single unified manner. The machine translation system used was a system comprised of two phrase-based SMT decoders. The first generated from the first token of the target to the last. The second system generated the target from last to first. Our results show that if only one of these decoding strategies is to be chosen, the optimal choice depends on the languages involved, and that in general a combination of the two approaches is able to outperform either approach.

1 Introduction

It is possible to couch the task of machine transliteration as a task of machine translation. Both processes involve the transformation of sequences of tokens in one language into sequences of tokens in another language. The principle differences between the machine translation and language translation are:

- Transliteration does not normally require the re-ordering of tokens that are generated in the target
- The number of types (the vocabulary size) in both source and target languages is considerably less for the transliteration task

We take a statistical machine translation paradigm (Brown at al., 1991) as the basis for our systems. The work in this paper is related to the work of (Finch and Sumita, 2008) who also use SMT directly to transliterate.

We view the task of machine transliteration as a process of machine translation at the character level (Donoual and LePage, 2006). We use state of the art phrase-based statistical machine translation systems (Koehn et al., 2003) to perform the transliteration. By adopting this approach we were able to build systems for all of the language pairs in the shared task using precisely the same procedures. No modeling of the phonetics of either source or target language (Knight and Graehl, 1997) was necessary, since the approach is simply a direct transformation of sequences of tokens in the source language into sequences of tokens in the target.

2 Overview

Our approach differs from the approach of (Finch and Sumita, 2008) in that we decode bidirectional. In a typical statistical machine translation system the sequence of target tokens is generated in a left-to-right manner, by left-to-right here we mean the target sequence is generated from the first token to its last. During the generation process the models (in particular the target language model) are able to refer to only the target tokens that have already been generated. In our approach, by using decoders that decode in both directions we are able to exploit context to the left and to the right of target tokens being generated. Furthermore, we expect our system to gain because it is a combination of two different MT systems that are performing the same task.

3 Experimental Conditions

In our experiments we used an in-house phrase-based statistical machine translation decoder called CleopATRa. This decoder operates on exactly the same principles as the publicly available MOSES decoder (Koehn et al., 2003). Like MOSES we utilize a future cost in our calculations. Our decoder was modified to be able to run two instances of the decoder at the same

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 52–56,
Suntec, Singapore, 7 August 2009.

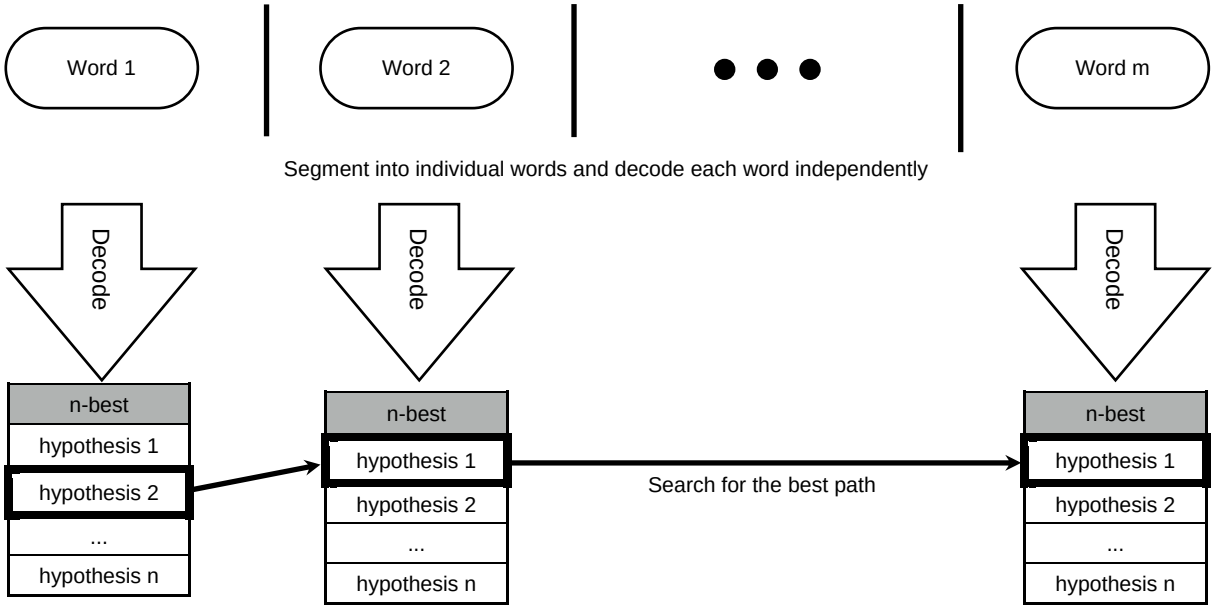

Figure 1: The decoding process for multi-word sequences

time. One instance decoding from left-to-right the other decoding from right-to-left. The hypotheses being combined by linearly interpolating the scores from both decoders at the end of the decoding process. In addition, the decoders were constrained decode in a monotone manner. That is, they were not allowed to re-order the phrases during decoding. The decoders were also configured to produce a list of unique sequences of tokens in their *n*-best lists. During SMT decoding it is possible to derive the same sequence of tokens in multiple ways. Multiply occurring sequences of this form were combined into a single hypothesis in the *n*-best list by summing their scores.

3.1 Pre-processing

In order to reduce data sparseness issues we took the decision to work with data in only its lowercase form. The only target language with case information was Russian. During the parameter tuning phase (where output translations are compared against a set of references) we restored the case for Russian by simply capitalizing the first character of each word.

We chose not to perform any tokenization for any of the language pairs in the shared task. We chose this approach for several reasons:

- It allowed us to have a single unified approach for all language pairs
- It was in the spirit of the evaluation, as it did not require specialist knowledge outside of the supplied corpora
- It enabled us to handle the Chinese names that occurred in the Japanese Romaji-Japanese Kanji task

However we believe that a more effective approach for Japanese-Kanji task may have been to re-tokenize the alphabetic characters into kana (for example transforming “k a” into the kana consonant vowel pair “ka”) since these are the basic building blocks of the Japanese language.

3.2 Training

For the final submission, all systems were trained on the union of the training data and development data. It was felt that the training set was sufficiently small that the inclusion of the development data into the training set would yield a reasonable boost in performance by increasing the coverage of the language model and phrase table. The language models and translation models were therefore built from all the data, and the log-linear weights used to combine the models of the systems were tuned using systems trained only on the training data. The development data in this case being held-out. It was assumed that these parameters would perform well in the systems trained on the combined development/training corpora.

3.3 Parameter Tuning

The SMT systems were tuned using the minimum error rate training procedure introduced in (Och, 2003). For convenience, we used BLEU as a proxy for the various metrics used in the shared task evaluation. The BLEU score is

	En-Ch	En-Ja	En-Ko	En-Ru	Jn-Jk
After tuning	0.908	0.772	0.622	0.914	0.769
Before tuning	0.871	0.635	0.543	0.832	0.737

Table 1: The effect on 1-best accuracy by tuning with respect to BLEU score

commonly used to evaluate the performance of machine translation systems and is a function of the geometric mean of *n*-gram precision. Table 1 shows the effect of tuning for BLEU on the ACC (1-best accuracy) scores for several languages. Improvements in the BLEU score also gave improvements in ACC. Tuning to maximize the BLEU score gave improvements for all language pairs and in all of the evaluation metrics used in this shared task. Nonetheless, it is reasonable to assume that one would be able to improve the performance in a particular evaluation metric by doing minimum error rate training specifically for that metric.

3.3.1 Multi-word sequences

The data for some languages (for example Hindi) contained some multi-word sequences. These posed a challenge for our approach, and gave us the following alternatives:

- Introduce a <space> token into the sequence, and treat it as one long character sequence to transliterate; or
- Segment the word sequences into individual words and transliterate these independently, combining the n-best hypothesis lists for all the individual words in the sequence into a single output sequence.

We adopted both approaches for the training of our systems. For those multi-word sequences where the number of words in the source and target matched, the latter approach was taken. For those where the numbers of source and target words differed, the former approach was taken. The decoding process for multi-word sequences is shown in Figure 1. This approach was only used during the parameter tuning on the development set, and in experiments to evaluate the system performance on development data since no multi-word sequences occurred in the test data.

During recombination, the score for the target word sequence was calculated as the product of the scores of each hypothesis for each word. Therefore a search over all combinations of hypotheses was required. In almost all cases we were able to perform a full search. For the rare long word sequences in the data, a beam search strategy was adopted.

3.3.2 Bidirectional Decoding

In SMT it is usual to decode generating the target sequence in order from the first token to the last token (we refer to this as left-to-right decoding, as this is the usual term for this, even though it may be confusing as some languages are naturally written from right-to-left). Since the decoding process is symmetrical, it is also possible to reverse the decoding process, generating from the end of the target sequence to the start (we will refer to this as right-to-left decoding). This reverse decoding is counter-intuitive since language is generated in a left-to-right manner by humans (by definition), however, in pilot experiments on language translation, we found that the best decoding strategy varies depending on the languages involved. The analogue of this observation was observed in our transliteration results (Table 1). For some language pairs, a left-to-right decoding strategy performed better, and for other language pairs the right-to-left strategy was preferable.

Our pilot experiments also showed that combining the hypotheses from both decoding processes almost always gave better results that the best of either left-to-right or right-to-left decoding. We observe a similar effect in the experiments presented here, although our results here are less consistent. This is possibly due to the differences in the size of the data sets used for the experiments. The data used in the experiments here being an order of magnitude smaller.

4 Results

The results of our experiments are shown in Table 1. These results are from a closed evaluation on development data. Only the training data were used to build the system's models, the development data being used to tune the log-linear weights for the translation engines' models and for evaluation. We show results for the case of equal interpolation weights of the left-to-right and right-to-left decoders. For the final submis-

Language	Decoding Strategy	ACC	Mean F-score	MRR	MAP_ref	MAP_10	MAP_sys
En-Ch	⟹	0.908	0.972	0.908	0.266	0.266	0.908
	⟸	0.914	0.974	0.914	0.268	0.268	0.914
	⟺	0.915	0.974	0.915	0.268	0.268	0.915
En-Hi	⟹	0.788	0.969	0.788	0.231	0.231	0.788
	⟸	0.785	0.968	0.785	0.230	0.230	0.785
	⟺	0.790	0.970	0.790	0.231	0.231	0.790
En-Ja	⟹	0.773	0.950	0.793	0.251	0.251	0.776
	⟸	0.767	0.948	0.785	0.249	0.249	0.768
	⟺	0.769	0.949	0.789	0.250	0.250	0.771
En-Ka	⟹	0.682	0.954	0.684	0.202	0.202	0.683
	⟸	0.660	0.953	0.661	0.195	0.195	0.660
	⟺	0.674	0.955	0.675	0.199	0.199	0.674
En-Ko	⟹	0.622	0.850	0.623	0.183	0.183	0.622
	⟸	0.620	0.851	0.621	0.182	0.182	0.619
	⟺	0.627	0.853	0.628	0.184	0.184	0.626
En-Ru	⟹	0.915	0.982	0.915	0.268	0.268	0.915
	⟸	0.921	0.983	0.921	0.270	0.270	0.921
	⟺	0.922	0.983	0.922	0.270	0.270	0.922
En-Ta	⟹	0.731	0.963	0.732	0.216	0.216	0.731
	⟸	0.734	0.962	0.735	0.217	0.217	0.735
	⟺	0.748	0.965	0.749	0.221	0.221	0.749
Jn-Jk	⟹	0.769	0.869	0.797	0.301	0.301	0.766
	⟸	0.766	0.862	0.792	0.299	0.299	0.761
	⟺	0.772	0.867	0.799	0.300	0.300	0.767

Table 2: Results showing the peformance of three decoding strategies with respect to the evaluation metrics used for the shared task. Here ⟹ denotes left-to-right decoding, ⟸ denotes right-to-left decoding and ⟺ denotes bidirectional decoding.

Key to Language Acronyms: En = English, Ch = Chinese, Hi = Hindi, Ja = Japanese Katakana, Ka = Kannada, Ko = Korean, Ru = Russian, Ta = Tamil, Jn = Japanese Romaji, Jk = Japanese Kanji.

sion these weights were tuned on the development data. The bidirectional performance was the best strategy for all but En-Ja and En-Ka in terms of ACC. This varies for other metrics but in general the bidirectional system most often gave the highest performance.

5 Conclusion

Our results show the performance of state of the art phrase-based machine translation techniques on the task of transliteration. We show that it is reasonable to use the BLEU score to tune the system, and that bidirectional decoding can improve performance. In future work we would like to consider more tightly coupling the decoders, introducing monotonicity into the alignment process, and adding contextual features into the translation models.

Acknowledgements

The results presented in this paper draw on the following data sets. For Chinese-English, Li et al., 2004. For Japanese-English, Korean-English, and Japanese(romaji)-Japanese(kanji), the reader is referred to the CJK website: http://www.cjk.org. For Hindi-English, Tamil-English, Kannada-English and Russian-English the data sets originated from the work of Kuraman and Kellner, 2007.

References

Peter Brown, S. Della Pietra, V. Della Pietra, and R. Mercer (1991). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2), 263-311.

Etienne Denoual and Yves Lepage. 2006. The character as an appropriate unit of processing for non-

segmenting languages, *Proceedings of the 12th Annual Meeting of The Association of NLP*, pp. 731-734.

Kevin Knight and Jonathan Graehl. 1997. Machine Transliteration. *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 128-135, Somerset, New Jersey.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. *In Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*, Edmonton, Canada.

Franz Josef Och, “Minimum error rate training for statistical machine translation,” *Proceedings of the ACL*, 2003.

Kumaran A., Kellner T., "A generic framework for machine transliteration", *Proc. of the 30th SIGIR*, 2007

Haizhou Li, Min Zhang, Jian Su, English-Chinese (EnCh): "A joint source channel model for machine transliteration", *Proc. of the 42nd ACL*, 2004.