Rescoring a Phrase-based Machine Transliteration System with Recurrent Neural Network Language Models

Andrew Finch
NICT
3-5 Hikaridai
Keihanna Science City
619-0289 JAPAN
andrew.finch@nict.go.jp

Paul Dixon
NICT
3-5 Hikaridai
Keihanna Science City
619-0289 JAPAN
paul.dixon@nict.go.jp

Eiichiro Sumita
NICT
3-5 Hikaridai
Keihanna Science City
619-0289 JAPAN
eiichiro.sumita@nict.go.jp

Abstract

The system entered into this year's shared transliteration evaluation is implemented within a phrase-based statistical machine transliteration (SMT) framework. The system is based on a joint source-channel model in combination with a target language model and models to control the length of the sequences generated. The joint source-channel model was trained using a many-to-many Bayesian bilingual alignment. The focus of this year's system is on input representation. In order attempt to mitigate data sparseness issues in the joint source-channel model, we augmented the system with recurrent neural network (RNN) models that can learn to project the grapheme set onto a smaller hidden representation. We performed experiments on development data to evaluate the effectiveness of our approach. Our results show that using an RNN language model can improve performance for language pairs with large grapheme sets on the target side.

1 Introduction

Our system for the NEWS shared evaluation on transliteration generation is based on the system entered into last years evaluation (Finch et al., 2011). Some minor improvements have been made to some of the components, but the major difference is the addition of a re-scoring step with three rescoring models: an RNN target language model; an RNN joint source-channel model; and a maximum entropy model (this model was part of last year's system but has been moved from the decoding step into the re-scoring step for efficiency). In all our experiments we have taken a strictly language independent approach. Each of the language pairs were processed automatically from the graphemic representation supplied for the shared tasks, with no language specific treatment for any of the language pairs.

Recent research results on the application of recurrent neural network models to language modeling have shown that very promising reductions in text data perplexity relative to traditional n-gram language model approaches are possible (Mikolov et al., 2010; Mikolov et al., 2011). The RNN approach differs from the standard n-gram approach in that RNNs are able to smooth by projecting the grapheme set onto a set of hidden units, a process that effectively clusters similar graphemes. Furthermore, RNNs have been reported to be effective where data resources are limited (Kombrink et al., 2011).

These characteristics motivate us to investigate the effect of applying this approach in modeling at the grapheme (or grapheme sequence pair) level, particularly as two of the most important models in our system are both language models. The main drawback of RNN based models, their exceptionally high training computational complexity (Mikolov et al., 2010) is not an obstacle for training models for this shared task, though it may be an issue if large amounts of monolingual data are used to build the language models. We run experiments using this technique to investigate its effect on both corpus perplexity and end-to-end system performance (since it is not necessarily the case that gains in language model perplexity result in better systems (Chen et al., 1998)).

Throughout this paper we will refer to graphemes, grapheme sequences and grapheme sequence pairs. By grapheme, we mean a single unicode character, for example 'a' in English, 'ア' in Japanese or '明' in Chinese. Grapheme sequences are arbitrary sequences of these graphemes, and grapheme sequence pairs are 2-tuples of grapheme sequences, each element in the tuple being a grapheme sequence in a given language; for example: ('hello','ハロー').

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 47–51,
Jeju, Republic of Korea, 8-14 July 2012.

2 System Description

2.1 Bilingual Bayesian Grapheme Alignment

To train the joint-source-channel model(s) in our system, we perform a many-to-many grapheme-to-grapheme alignment. To discover this alignment we use the Bayesian non-parametric technique described in (Finch and Sumita, 2010) which is a relative of the technique proposed by (Huang et al., 2011). Bayesian techniques typically build compact models with few parameters that do not overfit the data and have been shown to be effective for transliteration (Finch and Sumita, 2010; Finch et al., 2011).

2.2 Phrase-based SMT Models

The decoding was performed using a specially modified version of the OCTAVIAN decoder (Finch et al., 2007), an in-house multi-stack phrase-based decoder that operates on the same principles as the MOSES decoder (Koehn et al., 2007). This component of the system is implemented as a log-linear combination of 4 different models: a joint source-channel model; a target language model; a grapheme insertion penalty mode; and a grapheme sequence pair insertion penalty model. The following sections describe each of these models in detail. Due to the small size of many of the data sets in the shared tasks, we used all of the data to build models for the final systems.

2.2.1 N-gram joint source-channel model

The n-gram joint source-channel model used during decoding by the SMT decoder was trained from the Viterbi alignment arising from the final iteration of the Bayesian segmentation process on the training data (for the model used in parameter tuning), and the training data added to the development data (for the model used to decode the test data). We used the MIT language modeling toolkit (Bo-june et al., 2008) with modified Knesser-Ney smoothing to build this model. In all experiments we used a language model of order 5.

2.2.2 N-gram target Language model

The target model was trained from target side of the training data (for model used in parameter tuning), and the training data added to the development data (for the model used to decode the test data). We used the MIT language modeling toolkit with Knesser-Ney smoothing to build this model. In all experiments we used a language model of order 5.

2.2.3 Insertion penalty models

Both grapheme based and grapheme-sequence-pair-based insertion penalty models are simple models that add a constant value to their score each time a grapheme (or grapheme sequence pair) is added to the target hypotheses. These models control the tendency both of the joint source-channel model and the target language model to generate derivations that are too short.

2.3 Re-scoring Step

2.3.1 Overview

The system has a separate re-scoring stage that like the SMT models described in the previous section is implemented as a log-linear model. The log-linear weights are trained using the same MERT (Och, 2003) procedure. In principle, the weights for the models in this stage could be trained in a single step together with the SMT weights, and in last year's system this was the case for the ME model. However the models in this stage are more computationally expensive, and to reduce training time we train their weights in a second step. The three models used for re-scoring (20-best) are described in the following sections.

2.3.2 Maximum-entropy model

The maximum entropy model used for re-scoring embodies a set of features designed to take the local context of source and target graphemes and grapheme sequences into account. The features can be partitioned into two classes: grapheme-based features and grapheme sequence-based features. In both cases we use a context of 2 to the left and right for the source, and 2 to the left for the target. Sequence begin and end markers are added to both source and target and are used in the context. The features used in the ME model consist of all possible bigrams of contiguous elements in the context. We do not mix features at the grapheme level and grapheme sequence level, so for example, a grapheme sequence bigram can only consist of grapheme sequences (including sequences of length 1).

2.3.3 RNN Language models

We introduce two RNN language models (Mikolov et al., 2011) into the re-scoring step of our system. The first model is a language model over grapheme sequences in the target language; the second model is a joint source-channel model over bilingual grapheme sequence pairs. These models were trained on the same data as their

Language Pair	Accuracy in top-1	Mean F-score	MRR	MAP_{ref}
Arabic to English (ArEn)	0.588	0.930	0.709	0.507
Chinese to English (ChEn)	0.203	0.736	0.309	0.200
English to Bengali (Bangla) (EnBa)	0.460	0.891	0.583	0.458
English to Chinese (EnCh)	0.311	0.666	0.447	0.308
English to Hebrew (EnHe)	0.154	0.787	0.229	0.153
English to Hindi (EnHi)	0.668	0.923	0.738	0.661
English to Japanese Katakana (EnJa)	0.401	0.810	0.523	0.397
English to Kannada (EnKa)	0.546	0.901	0.641	0.545
English to Korean Hangul (EnKo)	0.384	0.721	0.465	0.383
English to Persian (EnPe)	0.655	0.941	0.774	0.643
English to Tamil (EnTa)	0.592	0.908	0.679	0.592
English to Thai (EnTh)	0.122	0.747	0.183	0.122
English to Japanese Kanji (JnJk)	0.513	0.693	0.598	0.419
Thai to English (ThEn)	0.140	0.766	0.216	0.140

Table 1: The evaluation results on the 2012 shared task for our system in terms of the official metrics.

n-gram counterparts described in Sections 2.2.1 and 2.2.2. The models were trained using the training procedure described in Section 3.1.

2.4 Parameter Tuning

The exponential log-linear model weights of both the SMT and re-scoring stages of our system were set by tuning the system on development data using the MERT procedure (Och, 2003) by means of the publicly available ZMERT toolkit [1] (Zaidan, 2009). The systems reported in this paper used a metric based on the word-level F-score, an official evaluation metric for the shared tasks (Zhang et al., 2012), which measures the relationship of the longest common subsequence of the transliteration pair to the lengths of both source and target sequences.

2.5 Official Results

The official scores for our system are given in Table 1. Some of the data tracks will benefit from a language-dependent treatment for example in Korean it is advantageous to decompose the characters, and other languages benefit from romanization as this can reduce data sparseness issue and allow the translation of unknown graphemes in test data.

3 Experiments

3.1 Perplexity

In this section we examine the performance of the RNN language model in terms of its perplexity on unseen data. For these experiments we divided the training into two parts: a training set (90% of the data) and a validation set (the remaining 10%), and used the development set as the test data on which the perplexity calculations were made.

The RNN model was built using the publicly available RNNLM toolkit [2]. A set of pilot experiments was run on subsets of the training data to find suitable values for the number of hidden units and number of classes used to train the RNN, and a simple grid search we used to find the best parameters for each language pair. All other parameters were left at their default values. The n-gram language model was trained using the SRI language modeling toolkit (Stolcke, 1999). We used a 5-gram model in these experiments trained with Witten-Bell smoothing.

Table 2 shows the results of this experiment. In 9 out of the 15 experiments the RNN language model had lower perplexity than the 5-gram backoff language model. Furthermore, in all of the experiments the interpolated model (a model formed by linearly interpolating the two models together with equal weights) had considerably lower perplexity than either component model. The largest relative gains were observed in Jn-Jk, En-Ko and En-Ch; these three languages had by far the largest grapheme set sizes out of all the language pairs. This result is not surprising because of the manner in which the RNN language models are able to smooth by projection of the grapheme set onto the set of hidden units.

[1] http://www.cs.jhu.edu/∼ozaidan/zmert/

[2] http://www.fit.vutbr.cz/ĩmikolov/rnnlm/index.html

Language Pair	RNN perplexity	N-gram perplexity	Interpolated perplexity	Grapheme set size	Corpus size (graphemes)	F-score with RNN	F-score no RNN
Ar-En	9.96	8.83	8.69	29	1683K	0.873	0.870
Ch-En	13.52	13.87	12.34	26	231K	0.896	0.882
En-Ba	12.30	11.00	10.73	59	78K	0.968	0.951
En-Ch	61.78	77.78	59.95	367	107K	0.883	0.866
En-He	9.78	10.27	9.51	34	49K	0.965	0.967
En-Hi	15.09	14.82	13.48	79	94K	0.980	0.977
En-Ja	19.52	20.16	18.51	81	132K	0.945	0.939
En-Ka	11.97	12.30	11.04	75	87K	0.967	0.969
En-Ko	45.06	50.41	44.79	700	19K	0.910	0.898
En-Pe	10.86	11.55	10.58	32	64K	0.933	0.937
En-Ta	9.23	9.49	8.60	63	93K	0.978	0.977
En-Th	8.40	8.23	7.67	64	207K	0.957	0.940
Jn-Jk	65.63	90.17	66.43	1536	44K	0.703	0.684
Th-En	10.20	9.37	8.98	43	166K	0.954	0.949

Table 2: Language model perplexity scores on the development set with n-gram, RNN and interpolated language models, together with system performance with and without the RNN models.

3.2 System Performance

In this section we look at whether the gains from incorporating the RNN language models result in gains in overall system performance. We ran experiments on the data used in the perplexity experiments. The only difference in the systems we compare was whether or not the RNN language models were included in the re-scoring process; the RNN model being effectively interpolated in a log-linear manner with the other models when it was included. MERT parameter tuning was performed separately for systems with and without the RNN models. The results in terms of F-score are shown in Table 2. The results show small gains in performance for 11 of the 14 language pairs, indicating that the RNN models are effective. Of the languages with larger grapheme set sizes that showed higher improvements in perplexity, two (Jn-Jk and En-Ch) showed larger than average improvement in overall system performance.

4 Conclusion

The system used for this year's shared evaluation was implemented within a phrase-based statistical machine translation framework augmented by a joint-source channel model trained from a many-to-many alignment of grapheme sequences using a Bayesian alignment approach. The system had a re-scoring step that integrates features from a maximum entropy model with two RNN language models; one for the target grapheme sequence, and the other for the sequence of grapheme sequence pairs used to generate the target.

We ran experiments to determine the effectiveness of the RNN language models on the transliteration tasks. We found that the approach was generally effective and particularly effective for tasks with large grapheme set sizes.

In future work we would like to investigate alternative ways of integrating RNN models into our system. In particular it may be feasible to insert the models directly into the SMT component of our system so that they can be used directly in the decoding process. Furthermore, we intend to examine how the impact of these models in the case where larger corpora of monolingual data are used.

Acknowledgements

For the English-Japanese, English-Korean and Arabic-English datasets, the reader is referred to the CJK website: http://www.cjk.org. For English-Hindi, English-Tamil, and English-Kannada, and English-Bangla the data sets originated from the work of (Kumaran and Kellner, 2007) [3]. The Chinese language corpora came from the Xinhua news agency (Xinhua News Agency, 1992). The English Persian corpus originates from the work of (Karimi et al., 2006; Karimi et al., 2007).

[3] http://research.microsoft.com/india

References

Bo-june, Paul Hsu, and James Glass. 2008. Iterative language model estimation: Efficient data structure and algorithms. In *Proc. Interspeech*.

Stanley Chen, Douglas Beeferman, and Ronald Rosenfeld. 1998. Evaluation metrics for language models.

Andrew Finch and Eiichiro Sumita. 2010. A Bayesian Model of Bilingual Segmentation for Transliteration. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 259–266.

Andrew Finch, Etienne Denoual, Hideo Okuma, Michael Paul, Hirofumi Yamamoto, Keiji Yasuda, Ruiqiang Zhang, and Eiichiro Sumita. 2007. The NICT/ATR speech translation system for IWSLT 2007. In *Proceedings of the IWSLT*, Trento, Italy.

Andrew Finch, Paul Dixon, and Eiichiro Sumita. 2011. Integrating models derived from non-parametric bayesian co-segmentation into a statistical machine transliteration system. In *Proceedings of the Named Entities Workshop*, pages 23–27, Chiang Mai, Thailand, Nov. Asian Federation of Natural Language Processing.

Yun Huang, Min Zhang, and Chew Lim Tan. 2011. Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars. In *ACL (Short Papers)*, pages 534–539.

Sarvnaz Karimi, Andrew Turpin, and Falk Scholer. 2006. English to persian transliteration. In *SPIRE*, pages 255–266.

Sarvnaz Karimi, Andrew Turpin, and Falk Scholer. 2007. Corpus effects on the evaluation of automated transliteration systems. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowa, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL 2007: proceedings of demo and poster sessions*, pages 177–180, Prague, Czeck Republic, June.

Stefan Kombrink, Tomáš Mikolov, Martin Karafiát, and Lukáš Burget. 2011. Recurrent neural network based language modeling in meeting recognition. In *Proceedings of Interspeech 2011*, volume 2011, pages 2877–2880. International Speech Communication Association.

A. Kumaran and Tobias Kellner. 2007. A generic framework for machine transliteration. In *SIGIR'07*, pages 721–722.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, volume 2010, pages 1045–1048. International Speech Communication Association.

Tomáš Mikolov, Anoop Deoras, Stefan Kombrink, Lukáš Burget, and Jan Černocký. 2011. Empirical evaluation and combination of advanced language modeling techniques. In *Proceedings of Interspeech 2011*, volume 2011, pages 605–608. International Speech Communication Association.

Franz J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the ACL*.

Andreas Stolcke. 1999. Srilm - an extensible language model toolkit.

Xinhua News Agency. 1992. Chinese transliteration of foreign personal names. *The Commercial Press*.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

Min Zhang, Haizhou Li, Liu Ming, and A. Kumaran. 2012. Whitepaper of news 2012 shared task on machine transliteration. In *Proceedings of the 2012 Named Entities Workshop*, Jeju, Korea. Association for Computational Linguistics.