Neural Network Transduction Models in Transliteration Generation

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

Lemao Liu
NICT
3-5 Hikaridai
Keihanna Science City
619-0289 JAPAN
lmliu@nict.go.jp

Xiaolin Wang
NICT
3-5 Hikaridai
Keihanna Science City
619-0289 JAPAN
xiaolin.wang@nict.go.jp

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

Abstract

In this paper we examine the effectiveness of neural network sequence-to-sequence transduction in the task of transliteration generation. In this year's shared evaluation we submitted two systems into all tasks. The primary system was based on the system used for the NEWS 2012 workshop, but was augmented with an additional feature which was the generation probability from a neural network. The secondary system was the neural network model used on its own together with a simple beam search algorithm. Our results show that adding the neural network score as a feature into the phrase-based statistical machine transliteration system was able to increase the performance of the system. In addition, although the neural network alone was not able to match the performance of our primary system (which exploits it), it was able to deliver a respectable performance for most language pairs which is very promising considering the recency of this technique.

1 Introduction

Our primary system for the NEWS shared evaluation on transliteration generation is based on the system entered into the 2012 evaluation (Finch et al., 2012) which in turn was a development of the 2011 system (Finch et al., 2011).

The system is based around the application of phrase-based statistical machine translation (PBSMT) techniques to the task of transliteration, as in (Finch and Sumita, 2008). The system differs from a typical phrase-based machine translation system in a number of important respects:

- Characters rather than words are used as the atomic elements used in the transductive process
- The generative process is constrained to be monotonic. No re-ordering model is used.
- The alignment process is constrained to be monotonic.
  - A non-parametric Bayesian aligner is used instead of GIZA++ and extraction heuristics, to provide a joint alignment/phrase pair induction process.
- The log-linear weights are tuned towards the F-score evaluation metric used in the NEWS evaluation, rather than a machine translation oriented score such as BLEU (Papineni et al., 2001).
- A bilingual language model (Li et al., 2004) is used as a feature during decoding.

An n-best list of hypotheses from the PBSMT system outlined above was then re-scored using the following set of models:

- A maximum entropy model (described in detail in (Finch et al., 2011)).

Proceedings of the Fifth Named Entity Workshop, joint with 53rd ACL and the 7th IJCNLP, pages 61–66,
Beijing, China, July 26-31, 2015.

- A recurrent neural network RNN target language model (Mikolov et al., 2010).
- An RNN bilingual language model (as in (Finch et al., 2012)).
- A neural network transliteration model (Bahdanau et al., 2014).

The re-scoring was done by extending the log-linear model of the PBSMT system with these 4 additional features. The weights for these features were tuned to maximize F-score in a second tuning step.

The novel aspect of our system in this year's evaluation is the use of a neural network that is capable of performing the entire transductive process. Neural networks capable of sequence-to-sequence transduction where the sequences are of different lengths (Hermann and Blunsom, 2013; Cho et al., 2014a; Bahdanau et al., 2014) are a very recent development in the field of machine translation. We believe this type of approach ought to be well suited to the task of transliteration, which is a task strongly related to that of machine translation but with typically much smaller vocabulary sizes and no problems related to reordering and in most cases no issues relating to out of vocabulary words (characters in our case). On the other hand, it is generally believed (for example (Ellis and Morgan, 1999)) that neural networks can require large amounts of data in order to train effective models, and the data set sizes available in this shared evaluation are quite small, and this lack of data may have caused problems for the neural networks employed.

In all our experiments we have taken a strictly language independent approach. Each of the language pairs were processed automatically from the character sequence representation supplied for the shared tasks, with no language specific treatment for any of the language pairs.

2 System Description

2.1 Non-parametric Bayesian Alignment

To train the joint-source-channel model(s) in our system, we perform a many-to-many sequence alignment. To discover this alignment we use the Bayesian non-parametric technique described in (Finch and Sumita, 2010). 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. The PBSMT component of the system was implemented as a log-linear combination of 4 different models: a joint source-channel model; a target language model; a character insertion penalty mode; and a character 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 (30 iterations were used) of the Bayesian segmentation process on the training data. We used the MIT language modeling toolkit (Bo-june et al., 2008) with modified Knesser-Ney smoothing to build this 5-gram model.

2.2.2 N-gram target Language model

The target language model was trained on the target side of the training data. We used the MIT language modeling toolkit with Knesser-Ney smoothing to build this 5-gram model.

2.2.3 Insertion penalty models

Both character based and character-sequence-pair-based insertion penalty models are simple models that add a constant value to their score each time a character (or character 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 encourage 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 (Finch et al., 2011). However the models in this stage are computationally expensive, and to reduce training time we train their weights in a second step. The

four 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 character and character-sequence based features designed to take the local context of source and target characters and character sequences into account; the reader is referred to (Finch et al., 2011) for a full description of this model.

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 character sequences in the target language; the second model is a joint source-channel model over bilingual character sequence pairs. These models were trained on the same data as their n-gram counterparts described in Sections 2.2.1 and 2.2.2. The models were trained using the training procedure described in (Finch et al., 2012).

2.3.4 Neural network transliteration model

The neural network transliteration model was trained directly from the source and target sequences themselves. The model used in tuning was trained only on the training data set; the model used for the final submission was trained on all of the data. The neural network software was developed using the GroundHog neural machine translation toolkit (Cho et al., 2014b), built on top of Theano (Bergstra et al., 2010; Bastien et al., 2012). For all of the experiments we used the same neural network architecture which was the default architecture supplied with the toolkit. That is, we used networks of 1000 hidden units and used the RNNSearch technique reported in (Bahdanau et al., 2014). In a set of pilot experiments we evaluated a number of neural network models with fewer parameters on development data, under the hypothesis that these would be more suitable for the task of transliteration. However, the best results came from the default set of parameters, and therefore these were used in all runs. Due to the resources required to train the neural network models only a few experiments were able to be performed and only on the English-Katakana task. It may be the case that different architectures could lead to in significantly higher performance than the results we obtained, and this remains an area for future research. The neural networks were trained for 50,000 iterations based on the analysis of the convergence of the performance on development data of a network trained on the English-Katakana task. The models took from 1 to 9 days to train, depending on the language pair, on a single core of a Tesla K40 GPU.

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.

3 Evaluation Results

The official scores for our system are given in Table 1. It is interesting to compare the results of the 2012 system with the results from this year's primary submission on the 2012 test set, since these results show the effect of adding the neural network transliteration scores into the re-scorer. In 11 out of 14 of the runs, the system's performance was improved, and for some language pairs, notably En-He, En-Hi, En-Ka, En-Pe, En-Ta, En-Th, Th-En and Jn-Jk the improvement was substantial. The using the neural network model scores was ineffective for Ar-En, Ch-En and En-Ch. Ar-En was surprising as the training corpus size for this task was considerably larger than for any other task, and we expected this to benefit the neural network approach. Overall however, it is clear from the results that the neural network re-scoring was very effective and the effect was considerably greater than that from the RNN re-scoring models introduced in the 2012 system.

The results on the Jn-Jk task were surprising. The neural network transliteration system alone produced very low accuracy scores, but when used in combination with the PBSMT system gave a 9.7% increase in top-1 accuracy. One particular characteristic of this data set is the disparity in length between the sequences; kanji sequences were very short whereas the romanized form was much longer. Visual inspection of the output from

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

Language Pair		2012	Primary		Secondary	
		system	2012	2011	2012	2011
Arabic to English	(ArEn)	0.588	0.529	0.527	0.469	0.494
English to Bengali	(EnBa)	0.460	0.483	0.479	0.364	0.375
Chinese to English	(ChEn)	0.203	0.184	0.158	0.136	0.115
English to Chinese	(EnCh)	0.311	0.313	0.344	0.220	0.213
English to Hebrew	(EnHe)	0.154	0.179	0.609	0.163	0.558
English to Hindi	(EnHi)	0.668	0.696	0.474	0.641	0.410
English to Japanese Katakana	(EnJa)	0.401	0.407	0.412	0.338	0.399
English to Kannada	(EnKa)	0.546	0.562	0.412	0.546	0.360
English to Korean Hangul	(EnKo)	0.384	0.363	0.365	0.189	0.200
English to Persian	(EnPe)	0.655	0.697	0.360	0.565	0.329
English to Tamil	(EnTa)	0.592	0.626	0.474	0.584	0.406
English to Thai	(EnTh)	0.122	0.157	0.387	0.132	0.359
English to Japanese Kanji	(JnJk)	0.513	0.610	0.452	0.032	0.035
Thai to English	(ThEn)	0.140	0.154	0.277	0.129	0.178

Table 1: The evaluation results on the 2015 shared task for our systems in terms of the top-1 accuracy.

the direct neural network transliteration showed that the output sequences derived from the roman character sequences, but were too long. When integrated with the PBSMT system, output sequences of this form were not a problem as they were rarely generated as candidates for re-scoring.

We conducted two experiments in the reverse direction from Jk to Jn. The first was based on a neural network transliteration system from character to character in the same manner as the secondary submission. The second system was a neural network that transduced from character to character sequence. We used a 1-to-many sequence alignment induced by the Bayesian aligner to train this model. The character-to-character system had a top-1 accuracy of 0.245, the character-to-character sequence system had a top-1 accuracy of 0.305. These results indicate that the neural network is capable of generating long sequences from short sequences with reasonably high accuracy, and that there may be something to be gained by using phrasal units in the neural network transduction process, as was the case when moving from word-based models to phrase-based models in machine translation.

4 Conclusion

The system used for this year's shared evaluation was implemented within a phrase-based statistical machine translation framework augmented by a bilingual language model trained from a many-to-many alignment from a non-parametric Bayesian aligner. The system had a re-scoring step that integrated features from a maximum entropy model, a target RNN language model, a bilingual RNN language model, and a neural network transliteration model.

Our results showed that the neural network transliteration model was a very effective component in the re-scoring stage of our system that substantially improved the performance of our system over the 2012 system for most language pairs. Furthermore, the neural network transliterator was a capable system in its own right on most of the tasks, and equaled or exceeded the performance of our 2012 system on 3 language pairs. These results are particularly impressive considering that this line of research is relatively new, and we believe neural network transliteration models will have a bright future in this field.

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) [2]. 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).

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

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June. Oral Presentation.

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

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*.

Dan Ellis and Nelson Morgan. 1999. Size matters: An empirical study of neural network training for large vocabulary continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 2, pages 1013–1016. IEEE.

Andrew Finch and Eiichiro Sumita. 2008. Phrase-based machine transliteration. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP)*, volume 1, Hyderabad, India.

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.

Andrew Finch, Paul Dixon, and Eiichiro Sumita. 2012. Rescoring a phrase-based machine transliteration system with recurrent neural network language models. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, pages 47–51, Jeju, Korea, July. Association for Computational Linguistics.

Karl Moritz Hermann and Phil Blunsom. 2013. A simple model for learning multilingual compositional semantics. *CoRR*, abs/1312.6173.

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*.

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

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 159, Morristown, NJ, USA. Association for Computational Linguistics.

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)*, number 9, 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*, number 8, pages 605–608. International Speech Communication Association.

Franz J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL 2003)*, Sapporo, Japan.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2001. *Bleu: a Method for Automatic Evaluation of Machine Translation*. IBM Research Report rc22176 (w0109022), Thomas J. Watson Research Center.

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.