Neural Machine Translation Techniques for Named Entity Transliteration

Neural Machine Translation Techniques for Named Entity Transliteration

Roman Grundkiewicz and **Kenneth Heafield**
University of Edinburgh
10 Crichton St, Edinburgh EH8 9AB, Scotland
{rgrundki,kheafiel}@inf.ed.ac.uk

Abstract

Transliterating named entities from one language into another can be approached as neural machine translation (NMT) problem, for which we use deep attentional RNN encoder-decoder models. To build a strong transliteration system, we apply well-established techniques from NMT, such as dropout regularization, model ensembling, rescoring with right-to-left models, and back-translation. Our submission to the NEWS 2018 Shared Task on Named Entity Transliteration ranked first in several tracks.

1 Introduction

Transliteration of Named Entities (NEs) is defined as the phonetic translation of names across languages (Knight and Graehl, 1998). It is an important part of a number of natural language processing tasks, and machine translation in particular (Durrani et al., 2014; Sennrich et al., 2016c).

Machine transliteration can be approached as a sequence-to-sequence modeling problem (Finch et al., 2016; Ameur et al., 2017). In this work, we explore the Neural Machine Translation (NMT) approach based on an attentional RNN encoder-decoder neural network architecture (Sutskever et al., 2014), motivated by its successful application to other sequence-to-sequence tasks, such as grammatical error correction (Yuan and Briscoe, 2016), automatic post-editing (Junczys-Dowmunt and Grundkiewicz, 2016), sentence summarization (Chopra et al., 2016), or paraphrasing (Mallinson et al., 2017). We apply well-established techniques from NMT to machine transliteration building a strong system that achieves state-of-the-art-results. The techniques we exploit include:

- Regularization with various dropouts preventing model overfitting;
- Ensembling strategies involving independently trained models and model checkpoints;
- Re-scoring of n-best list of candidate transliterations by right-to-left models;
- Using synthetic training data generated via back-translation.

The developed system constitutes our submission to the NEWS 2018 Shared Task[1] on Named Entity Transliteration ranked first in several tracks.

We describe the shared task in Section 2, including provided data sets and evaluation metrics. In Section 3, we present the model architecture and adopted NMT techniques. The experiment details are presented in Section 4, the results are reported in Section 5, and we conclude in Section 6.

2 Shared task on named entity transliteration

The NEWS 2018 shared task (Chen et al., 2018) continues the tradition from the previous tasks (Xiangyu Duan et al., 2016, 2015; Zhang et al., 2012) and focuses on transliteration of personal and place names from English or into English or in both directions.

2.1 Datasets

Five different datasets have been made available for use as the training and development data. The data for Thai (EnTh, ThEn) comes from the NECTEC transliteration dataset. The second dataset is the RMIT English-Persian dataset (Karimi et al., 2006, 2007) (EnPe, PeEn). Chinese (EnCh, ChEn) and Vietnamese (EnVi) data originates in Xinhua

[1] http://workshop.colips.org/news2018

ID	Languages	Train	Dev	Test
EnTh	English-Thai	30,781	1000	1000
ThEn	Thai-English	27,273	1000	1000
EnPe	English-Persian	13,386	1000	1000
PeEn	Persian-English	15,677	1000	1000
EnCh	English-Chinese	41,318	1000	1000
ChEn	Chinese-English	32,002	1000	1000
EnVi	English-Vietnamese	3,256	500	500
EnHi	English-Hindi	12,937	1000	1000
EnTa	English-Tamil	10,957	1000	1000
EnKa	English-Kannada	10,955	1000	1000
EnBa	English-Bangla	13,623	1000	1000
EnHe	English-Hebrew	10,501	1000	1000
HeEn	Hebrew-English	9,447	1000	1000

Table 1: Official data sets in NEWS 2018 which we use in our experiments.

transliteration datasets (Haizhou et al., 2004), and the VNU-HCMUS dataset (Cao et al., 2010; Ngo et al., 2015), respectively. Hindi, Tamil, Kannada, Bangla (EnHi, EnTa, EnKa, EnBa), and Hebrew (EnHe, HeEn) are provided by Microsoft Research India[2]. We do not evaluate our models on the dataset from the CJK Dictionary Institute as the data is not freely available for research purposes.

We use 13 data sets for our experiments (Table 1). The data consists of genuine transliterations or back-translations or includes both.

No other parallel nor monolingual data are allowed for the constrained standard submissions that we participate in.

2.2 Evaluation

The quality of machine transliterations is evaluated with four automatic metrics in the shared task: word accuracy, mean F-score, mean reciprocal rank, and MAP_{ref} (Chen et al., 2018). As a main evaluation metric for our experiments we use word accuracy (Acc) on the top candidate:

$$Acc = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1 & \text{if } c_{i,1} \text{matches any of } r_{i,j}\\ 0 & \text{otherwise}\end{cases}.$$

The closer the value to 1.0, the more top candidates $c_{i,1}$ are correct transliterations, i.e. they match one of the references $r_{i,j}$. N is the total number of entries in a test set.

3 Neural machine translation

Our machine transliteration system is based on a deep RNN-based attentional encoder-decoder model that consists of a bidirectional multi-layer encoder and decoder, both using GRUs as their RNN variants (Sennrich et al., 2017b). It utilizes the BiDeep architecture proposed by Miceli Barone et al. (2017), which combines deep transitions with stacked RNNs. We employ the soft-attention mechanism (Bahdanau et al., 2014), and leave hard monotonic attention models (Aharoni and Goldberg, 2017) for future work. Layer normalization (Ba et al., 2016) is applied to all recurrent and feed-forward layers, except for layers followed by a softmax. We use weight tying between target and output embeddings (Press and Wolf, 2017).

The model operates on word level, and no special adaptation is made to the model architecture in order to support character-level transliteration, except data preprocessing (Section 4.1).

3.1 NMT techniques

Regularization Randomly dropping units from the neural network during training is an effective regularization method that prevents the model from overfitting (Srivastava et al., 2014).

For RNN networks, Gal and Ghahramani (2016) proposed variational dropout over RNN inputs and states, which we adopt in our experiments. Following Sennrich et al. (2016a), we also dropout entire source and target words (characters in our case) with a given probability.

Model ensembling Model ensembling leads to consistent improvements for NMT (Sutskever et al., 2014; Sennrich et al., 2016a; Denkowski and Neubig, 2017). An ensemble of independent models usually outperforms an ensemble of different model checkpoints from a single training run as it results in more diverse models in the ensemble (Sennrich et al., 2017a). As an alternative method for checkpoint ensembles, Junczys-Dowmunt et al. (2016) propose exponential smoothing of network parameters averaging them over the entire training.

We combine both methods and build ensembles of independently trained models with exponentially smoothed parameters.

Re-scoring with right-left models Re-scoring of an n-best list of candidate translations obtained from one system by another allows to incorporate additional features into the model or to combine multiple different systems that cannot be easily ensembled. Sennrich et al. (2016a, 2017a), for re-scoring a NMT system, propose to use separate

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

ID	Original	+Synthetic	R
EnTh	59,131	154,232	×1
ThEn	58,872	153,973	×1
EnPe	32,321	127,314	×1
PeEn	32,616	127,609	×1
EnCh	81,252	176,367	×1
ChEn	80,818	175,933	×1
EnVi	2,756	139,175	×16
EnHi	12,607	145,507	×4
EnTa	10,702	137,887	×4
EnKa	10,662	137,727	×4
EnBa	13,389	148,635	×4
EnHe	18,558	132,070	×2
HeEn	18,388	131,730	×2

Table 2: Comparison of training data sets without and with synthetic examples. The original data are oversampled R times in synthetic data sets.

models trained on reversed target side that produce the target text from right-to-left.

We adopt the following re-ranking technique: we first ensemble four standard left-to-right models to produce n-best lists of 20 transliteration candidates and then re-score them with two right-to-left models and re-rank.

Back-translation Monolingual data can be back-translated by a system trained on the reversed language direction to generate synthetic parallel corpora (Sennrich et al., 2016b). Additional training data can significantly improve a NMT system.

As the task is organized under a constrained settings and no data other than that provided by organizers is allowed, we consider the English examples from all datasets as our monolingual data and use back-translations and "forward-translations" to enlarge the amount of parallel training data.

4 Experimental setting

We train all systems with Marian NMT toolkit[3,4] (Junczys-Dowmunt et al., 2018).

4.1 Data preprocessing

We uppercase[5] and tokenize all words into sequences of characters and treat them as words. Whitespaces are replaced by a special character to be able to reconstruct word boundaries after decoding.

[3]https://marian-nmt.github.io

[4]The training scripts are available at http://github.com/snukky/news-translit-nmt.

[5]The evaluation metric is case-insensitive.

We use the training data provided in the NEWS 2018 shared task to create our training and validation sets, and the official development set as an internal test set. Validation sets consists of randomly selected 500 examples that are subtracted from the training data. If a name entity has alternative translations, we add them to the training data as separate examples with identical source side. The number of training examples varies between ca. 2,756 and 81,252 (Table 2).

4.2 Model architecture

We use the BiDeep model architecture (Miceli Barone et al., 2017) for all systems. The model consists of 4 bidirectional alternating stacked encoders with 2-layer transition cells, and 4 stacked decoders with the transition depth of 4 in the base RNN of the stack and 2 in the higher RNNs. We augment it with layer normalization, skip connections, and parameter tying between all embeddings and output layer. The RNN hidden state size is set to 1024, embeddings size to 512. Source and target vocabularies are identical. The size of the vocabulary varies across language pair and is determined by the number of unique characters in the training data.

4.3 Training settings

We limit the maximum input length to 80 characters during training. Variational dropout on all RNN inputs and states is set to 0.2, source and target dropouts are 0.1. A factor for exponential smoothing is set to 0.0001.

Optimization is performed with Adam (Kingma and Ba, 2014) with a mini-batch size fitted into 3GB of GPU memory[6]. Models are validated and saved every 500 mini-batches. We stop training when the cross-entropy cost on the validation set fails to reach a new minimum for 5 consecutive validation steps. As a final model we choose the one that achieves the highest word accuracy on the validation set. We train with learning rate of 0.003 and decrease the value by 0.9 every time the validation score does not improve over the current best value. We do not change any training hyperparameters across languages.

Decoding is done by beam search with a beam size of 10. The scores for each candidate translation are normalized by sentence length.

[6]We train all systems on a single GPU.

System	EnTh	ThEn	EnPe	PeEn	EnCh	ChEn	EnVi	EnHi	EnTa	EnKa	EnBa	EnHe	HeEn
No dropouts	0.434	0.467	0.566	0.365	0.754	0.306	0.390	0.466	0.451	0.387	0.450	0.616	0.286
Baseline model	0.467	0.503	0.594	0.390	0.739	0.347	0.458	0.481	0.455	0.418	0.465	0.632	0.284
Right-left model	0.462	0.502	0.598	0.402	0.751	0.351	0.458	0.476	0.446	0.403	0.476	0.606	0.287
Ensemble ×4	0.477	0.526	0.605	0.407	0.752	0.366	0.478	0.504	0.469	0.438	0.489	0.633	0.291
+ Re-ranking	0.475	0.534	0.606	0.436	**0.765**	0.365	0.494	0.515	**0.483**	0.441	**0.488**	**0.638**	0.294
+ Synthetic data	**0.484**	**0.728**	**0.610**	**0.585**	0.760	**0.759**	**0.496**	**0.519**	0.471	**0.455**	0.484	0.626	**0.615**
Test set	0.167	0.328	—	—	0.304	0.276	0.502	0.333	0.237	0.340	0.461	0.187	0.153

Table 3: Results (Acc) on the official NEWS 2018 development set. Bolded systems have been evaluated on the official test set (last row).

4.4 Synthetic parallel data

English texts from parallel training data from all datasets are used as monolingual data from which we generate synthetic examples[7]. We do not make a distinction between authentic examples or actual back-translations, and collect 95,179 unique English named entities in total.

We back-translate English examples using the systems trained on the original data and use them as additional training data for training the systems into English. For systems from English into another language, we translate English texts with analogous systems creating "forward-translations". To have a reasonable balance between synthetic and original examples, we oversample the original data several times (Table 2). The number of oversampling repetitions depends on the language pair, for instance, the Vietnamese original data are oversampled 16 times, while Chinese data are not oversampled at all.

5 Results on the development set

We evaluate our methods on the official development set from the NEWS 2018 shared task (Table 3). Results for systems that do not use ensembles are averaged scores from four models.

Regularization with dropouts improves the word accuracy for all language pairs except English-Chinese. As expected, model ensembling brings significant and consistent gains. Re-ranking with right-to-left models is also an effective method raising accuracy, even for languages for which a single right-to-left model itself is worse then a baseline left-to-right model, e.g. for EnHi, EnKa and EnHe systems.

The scale of the improvement for systems trained on additional synthetic data depends on the method that the synthetic examples are generated with: the systems into English benefit greatly from back-translations[8], while other systems that were supplied by forward-translations do not improve much or even slightly downgrade the accuracy.

6 Official results and conclusions

As final systems submitted to the NEWS 2018 shared task we chose ones that achieved the best performance on the development set (Table 3, last row). On the official test set, our systems are ranked first for most language pairs we experimented with[9].

The results show that the neural machine translation approach can be employed to build efficient machine transliteration systems achieving state-of-the-art results for multiple languages and providing strong baselines for future work.

Acknowledgments

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

[7] More specifically, we use the source side of EnTh, EnPe, EnCh, EnVi, EnHi, EnTa, EnKa, EnBa, EnHe, and the target side of ThEn, PeEn, ChEn, HeEn data sets.

[8] The part of improvements might come from the fact that the ThEn, PeEn, ChEn and HeEn data sets have been created via back-translations and may include some of the examples from the development set.

[9] Due to issues with the test set, at the time of the camera-ready preparation, there were no official results for Persian.

References

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

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2004–2015.

Hadj Ameur, Farid Meziane, Ahmed Guessoum, et al. 2017. Arabic machine transliteration using an attention-based encoder-decoder model. *Procedia Computer Science*, 117:287–297.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Nam X. Cao, Nhut M. Pham, and Quan H. Vu. 2010. Comparative analysis of transliteration techniques based on statistical machine translation and joint-sequence model. In *Proceedings of the 2010 Symposium on Information and Communication Technology, SoICT 2010, Hanoi, Viet Nam, August 27-28, 2010*, pages 59–63.

Nancy Chen, Xiangyu Duan, Min Zhang, Rafael Banchs, and Haizhou Li. 2018. Whitepaper of NEWS 2018 shared task on machine transliteration. In *Proceedings of the Seventh Named Entity Workshop*. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *The First Workshop on Neural Machine Translation (NMT)*, Vancouver, Canada.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an unsupervised transliteration model into statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 148–153.

Andrew Finch, Lemao Liu, Xiaolin Wang, and Eiichiro Sumita. 2016. Target-bidirectional neural models for machine transliteration. In *Proceedings of the Sixth Named Entity Workshop*, pages 78–82. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

Li Haizhou, Zhang Min, and Su Jian. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Rico Sennrich. 2016. The AMU-UEDIN submission to the WMT16 news translation task: Attention-based nmt models as feature functions in phrase-based SMT. In *Proceedings of the First Conference on Machine Translation*, pages 319–325, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 751–758.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++.

Sarvnaz Karimi, Andrew Turpin, and Falk Scholer. 2006. English to persian transliteration. In *String Processing and Information Retrieval, 13th International Conference, SPIRE 2006, Glasgow, UK, October 11-13, 2006, Proceedings*, pages 255–266.

Sarvnaz Karimi, Andrew Turpin, and Falk Scholer. 2007. Corpus effects on the evaluation of automated transliteration systems. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 881–893.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep Architectures for Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Hoang Gia Ngo, Nancy F. Chen, Binh Minh Nguyen, Bin Ma, and Haizhou Li. 2015. Phonology-augmented statistical transliteration for low-resource languages. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 3670–3674.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 157–163.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017a. The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, WMT 2017*, pages 389–399.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017b. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

China Xiangyu Duan, Soochow University, Singapore Rafael E Banchs, Institute for Infocomm Research, China Min Zhang, Soochow University, Singapore Haizhou Li, Institute for Infocomm Research, and India A Kumaran, Microsoft Research, editors. 2015. *Proceedings of the Fifth Named Entity Workshop*. Association for Computational Linguistics, Beijing, China.

China Xiangyu Duan, Soochow University, Singapore Rafael E Banchs, Institute for Infocomm Research, China Min Zhang, Soochow University, Singapore Haizhou Li, Institute for Infocomm Research, and India A Kumaran, Microsoft Research, editors. 2016. *Proceedings of the Sixth Named Entity Workshop*. Association for Computational Linguistics, Berlin, Germany.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386.

Min Zhang, Haizhou Li, and A Kumaran, editors. 2012. *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*. Association for Computational Linguistics, Jeju, Korea.