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 on Named Entity Transliteration ranked first in several tracks.

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

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

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.

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 alternating 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.

\(^3\)https://marian-nmt.github.io
\(^5\)The evaluation metric is case-insensitive.
\(^6\)We train all systems on a single GPU.
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. 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 than 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, 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.

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.

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

<table>
<thead>
<tr>
<th>System</th>
<th>EnTh</th>
<th>ThEn</th>
<th>EnPe</th>
<th>PeEn</th>
<th>EnCh</th>
<th>ChEn</th>
<th>EnVi</th>
<th>ViEn</th>
<th>EnHi</th>
<th>HiEn</th>
<th>EnTa</th>
<th>TaEn</th>
<th>EnKa</th>
<th>KaEn</th>
<th>EnBa</th>
<th>BaEn</th>
<th>EnHe</th>
<th>HeEn</th>
</tr>
</thead>
<tbody>
<tr>
<td>No dropouts</td>
<td>0.434</td>
<td>0.467</td>
<td>0.566</td>
<td>0.365</td>
<td>0.754</td>
<td>0.306</td>
<td>0.390</td>
<td></td>
<td>0.466</td>
<td>0.451</td>
<td>0.387</td>
<td>0.450</td>
<td>0.616</td>
<td>0.286</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline model</td>
<td>0.467</td>
<td>0.503</td>
<td>0.594</td>
<td>0.390</td>
<td>0.739</td>
<td>0.347</td>
<td>0.458</td>
<td></td>
<td>0.481</td>
<td>0.455</td>
<td>0.403</td>
<td>0.476</td>
<td>0.606</td>
<td>0.287</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right-left model</td>
<td>0.462</td>
<td>0.502</td>
<td>0.598</td>
<td>0.402</td>
<td>0.751</td>
<td>0.351</td>
<td>0.458</td>
<td></td>
<td>0.476</td>
<td>0.446</td>
<td>0.403</td>
<td>0.476</td>
<td>0.606</td>
<td>0.287</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensemble × 4</td>
<td>0.477</td>
<td>0.526</td>
<td>0.605</td>
<td>0.407</td>
<td>0.752</td>
<td>0.366</td>
<td>0.478</td>
<td></td>
<td>0.504</td>
<td>0.469</td>
<td>0.438</td>
<td>0.489</td>
<td>0.633</td>
<td>0.291</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ Re-ranking</td>
<td>0.475</td>
<td>0.534</td>
<td>0.606</td>
<td>0.436</td>
<td>0.765</td>
<td>0.365</td>
<td>0.494</td>
<td></td>
<td>0.515</td>
<td>0.483</td>
<td>0.441</td>
<td>0.488</td>
<td>0.638</td>
<td>0.294</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ Synthetic data</td>
<td>0.484</td>
<td>0.728</td>
<td>0.610</td>
<td>0.585</td>
<td>0.760</td>
<td>0.759</td>
<td>0.496</td>
<td></td>
<td>0.519</td>
<td>0.747</td>
<td>0.455</td>
<td>0.484</td>
<td>0.626</td>
<td>0.615</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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


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.
