A Training and hyperparameters

In this appendix we provide all the information required to reproduce our results. The models have been implemented by modifying OpenNMT (Klein et al., 2017) and we will release our code publicly immediately after the anonymity period. All the code is already available to the reviewers as supplementary material.

To build a strong and current baseline, we have closely followed the indications of (Denkowski and Neubig, 2017). The baseline uses a single-layer bidirectional LSTM and a unidirectional LSTM as encoder and decoder, respectively. The attention mechanism is that of (Bahdanau et al., 2015). We have set the size of the LSTMs’ hidden layer to 1024, the size of the attention layer to the same size, and the size of the word embeddings to 300. We have initialized the word embeddings with the publicly-available pre-trained vectors from fastText\textsuperscript{1} for each language. The maximum length of the training sentences has been set to 100 tokens. The model vocabulary has been limited to 50,000 words for both the source and target languages. Words that are not present in the vocabulary are mapped to an \textit{unk} token, but are later replaced with the corresponding source word with highest attention, following (Luong et al., 2015). For inference, we have used beam search with a beam size of 5.

We have added ReWE to this baseline, keeping all the aforementioned values unchanged. As mentioned in the paper, ReWE is a stack of two linear layers with a ReLU in between. The first linear layer reduces vector \( s_j \) from size 1024 to 200. After the ReLU, the second linear layer expands the vector from size 200 to 300, which is the size of the word embeddings. The value for \( \lambda \) has been selected by evaluating the model over the en-fr validation set (see Section 4.2 in the paper).

All the models have been trained until convergence of the perplexity, using the Adam optimizer (Kingma and Ba, 2015), with a maximum step size of 0.0002, multiple restarts, and learning rate annealing (Denkowski and Neubig, 2017). After three consecutive validation evaluations without perplexity improvement, we halve the learning rate, and we repeat this process 5 times. After the 5-th halving, we stop the training if there is no perplexity improvement over 20 consecutive runs. The batch size is 40 and the model is evaluated every 25,000 sentences.

We have also trained the models at sub-word level using byte pair encoding (BPE) (Sennrich et al., 2016). We have learned the sub-word models using the concatenated training sets of all datasets, setting the number of merge operations to 32,000 for en-fr and cs-en, and to 8,000 for eu-en, given its much smaller size. We have also pre-trained word embeddings of size 300 for the new sub-word vocabularies, and used them for initialization of the word embeddings.

For each model, we have reported the average BLEU score (Papineni et al., 2002) of 10 independent runs, except for the selection of \( \lambda \) where we have averaged only 3 independent runs.

B Translation examples

In this section we showcase more examples of translations made by the model with and without ReWE for all the language pairs evaluated in the paper (en-fr, cs-en and eu-en). In general the translations made by ReWE seem to preserve a higher amount of information from the original source sentence, which is often referred to as higher “adequacy”.

\textsuperscript{1}https://fasttext.cc/docs/en/crawl-vectors.html
C Constrastive experiments

To gain further insight on the performance of the proposed technique, we have added two contrastive experiments. The first one (Contrastive A) removes ReWE from the architecture, but still retains the combined loss function (Eq. 7 in the paper). Instead of computing the ReWELoss between the ground-truth embedding and the regressed embedding, we compute it between the
Figure 1: Plot of the values of various loss functions during training of our model over the en-fr training set: green, ●: training loss (NLL + (\(\lambda = 20\)) ReWE (MSE); Eq.7); red, +: NLL loss; blue, dashed: ReWE (MSE) loss; magenta, ×: ReWE (MSE) loss scaled by \(\lambda = 20\). Each point in the graph is an average value of the corresponding loss over 25,000 sentences.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BLEU</th>
<th>BPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-fr</td>
<td>33.82</td>
<td>33.37</td>
</tr>
<tr>
<td>cs-en</td>
<td>20.70</td>
<td>22.53</td>
</tr>
<tr>
<td>eu-en</td>
<td>12.15</td>
<td>17.53</td>
</tr>
</tbody>
</table>

Table 4: Results of the Contrastive A experiment (\(\lambda = 0.2\); average of 10 models trained independently from different random seeds).

This shows that, in comparison, the proposed joint learning is a much more effective setting.

In turn, the Contrastive B experiment has achieved much lower BLEU scores. The first experiment over the cs-en dataset reported only 12.71 BLEU points (average of 10 independent runs), approximately half of the other models. Due to this poor result, we have not carried out this experiment further. Our interpretation of this result is that targeting the word embedding is an effective regularizer in the continuous domain, but the conversion of the predicted word embedding to a categorical value is prone to errors from closer neighbors.

D Behaviour of the ReWE (MSE) loss

Figure 1 plots the values of the NLL and ReWE (MSE) losses during training of our model over the en-fr training set. The ReWE (MSE) loss shows large fluctuations as the training progresses, with major increases at the re-starts of the optimizer for the simulated annealing that are not compensated for by the rest of the training.

References


