11-731 Machine Translation

Assignment 2 Report

Jin Cao, Kaiyu Zheng, Xinyu Chang

https://github.com/WhiplashCxy/11731-Assignment2

1 Overview

In this assignment, we tried to achieve the challenging task of translating English into three low-resource languages using neural machine translation. Apart from the baseline implementation, we also tried several methods tailored to this task including back translation, teacher forcing and label smoothing. Finally, we achieved BLEU scores of 39.05, 39.94, 19.81 on en-af, en-ts, en-nso machine translation tasks, respectively.

2 Experiments

2.1 External Parallel Corpus

Since all three target languages in this assignment are low-resources, the limited size of training data would constraints the final result of our model. So we just integrated other extra resources into our training data. Following the guide listed in the piazza, we imported resources from OPUS and did some preprocessing for data by filtering out infrequent words. Other than the size of extra dataset, the quality of them should also be considered. Since one word with the same meaning could be expressed with different format, even with the same meaning, the different result would still cause low bad result. That is to say, the context of the training data is really important and wrong context, such as those are filled with many religious background in some dataset from OPUS, would still lead to a bad result.

2.2 Back Translation

The extra corpus is one way to solve the low-resource problem, another way for exploiting the monolingual training data is back translation. The method is to translate reversely from the target language to the source language. Then the result of translation could combined with original source input as new synthetic training data.

2.3 Transfer learning

Given that the three languages are all low-resources, we thought that transfer learning might be a good idea, since it can help to gain experience from trans-

lating other languages and therefore improve the performance. The other data set we used was **en_nl**. We first train our model on **en_nl** for 50 epochs, and then we fix the encoder, and train the model on three other datasets for another 50 epochs.

2.4 Teacher forcing learning

Noticed that in training, the target is fed as input, which is equivalent to using teacher forcing with ratio 1.0. We added teacher forcing learning in the transfer model. For training, we set the teacher force ratio 0.9. Because we need to feed the prediction as input into the model, so we need to iterate the sequence, which makes it really slow to train a model. Therefore, we only experiment teacher forcing learning on en_nso dataset.

2.5 Label smoothing

As a regularization technique, label smoothing is applied to prevent the model from predicting the training examples too confidently. Before applying label smoothing, we are using one-hot encoded labels for computing the loss. Now instead of using one-hot encoded vector, we introduce noise distribution

$$
\begin{aligned}
p'(y|x_i) &= (1-\varepsilon)p(y|x_i) + \varepsilon u(y|x_i) \\
&= \begin{cases} 1-\varepsilon+\varepsilon u(y|x_i) & \text{if } y = y_i \\ \varepsilon u(y|x_i) & \text{otherwise} \end{cases}
\end{aligned}
$$

3 Result and Discussion

3.1 Hyper-parameter

When we trained the model using the baseline script, we found that the model got overfitted very quickly. We think the reason was we are training the model on low-resource languages this time, the data set itself is pretty small, so we attempted to enlarge the dataset by lowering the dropout hyperparameter. After our experiments, it generally gives us better results when we set the dropout rate to a lower number than that in the baseline scripts. In the translation process, we increase the beam search size to 20 and the learning rate decay is implemented in the program. Other parameters are listed in below table.

n-layers	n-heads	embed-dim	hidden-dim	dropout	lr	n-epochs
4	4	512	512	0.1	5e-4	50

Table 1: Result for dev

	en-af	en-ts	en-nso
Label Smoothing	39.05	39.94	19.81
Teacher Forcing	-	-	17.25
Transfer Learning	33.21	32.48	17.12

3.2 Transfer learning

Different form other transfer learning where low resources languages are translated into English, we are translating English into low resources language. Such difference lead to the poor performance of transfer learning. Because decoder is hard to train compared with encoder, translating into English means we can well train the decoder by using other language resources. However, translating into low resources languages means we can only well train the encoder, but the decoder can't learn well on small data sets.

3.3 Result

Since the teacher forcing is really time-consuming and the result is worse than other model, we just stop applying this method to other dataset after getting the result for the en-nso. The results are list in the Table 1.

4 Team Work

Since our team did all work without git and only utilized it to submit our result, commits of repository are not clear to show tasks for team members. So we list the overall sub-tasks for each member.

- Jin Cao: Label smoothing, teacher forcing, preprocessing for parallel corpus and find tuned hyperparameters.
- Kaiyu Zheng: Back translation, preprocessing for parallel corpus, find tuned hyperparameters.
- Xinyu Chang: Transfer learning, preprocessing for parallel corpus, find tuned hyperparameters.