11731 Assignment-2 Report: Symbolic Machine Translation Model

Hongliang Yu, Keyang Xu, Shuxin Yao
Language Technologies Institute
{hongliay, keyangx, shuxiny}@andrew.cmu.edu

1 Overview

In this assignment, we created a German-English symbolic machine translation model with some modifications, including IBM 1, IBM 2, phrase extraction algorithm and WSFT construction. Compared with a vanilla model, we achieve significant improvements for BLEU scores on the IWSLT test and validation dataset.

The implementations can be view on our Github: `https://github.com/yuhongliang324/Seq2Seq-HW2`. The folder "src" places all our codes. And the "output" folder contains our results for official evaluation on "test" and "blind" sets.

2 Methods and Implementations

In this section, we will describe our specific implementations for the translation model. Open-source package OpenFST is utilized in constructing WFST.

2.1 IBM 1 Model

We implemented the basic IBM 1 model in `train_model1.py` script. The alignment probabilities are optimized using EM algorithm. In the E-step, the probability of a single alignment $P(a_j = t|f_j, E)$ is estimated using the following Bayesian's rule:

$$P(a_j = t|f_j, E) = \frac{P(f_j|a_j = t, E)}{\sum_{\hat{t}=1}^{|E|+1} P(f_j|a_j = \hat{t}, E)} \tag{1}$$

In the M-step, we calculate the parameters $\theta_{f,e}$. For German-English paired texts, we inspect the largest English alignments for each German token.

2.1.1 Phrase Extraction and Phrase Length Constraint

In the phrase extraction, we follow the Algorithm 6 in Lecture 13. There are two modifications:

(1) We filter the transition pairs with counting less than 2. By this way, the candidate pairs are dramatically decreased from more than 3 million to less than 0.2M.

(2) *Phrase Length Constraint*: We remain the phrases which have lengths no more than 3. The candidate pairs are decreased from 170K to 140K.

2.1.2 WFST

In the `create-phrase-fst.py`, we generate a WFST for the phrase-based translation model as Figure 41. For each phrase pair, we create a path through the WFST that reads in the words in source language one by one, and outputs the words in target language one by one, and finally adds the log probability by an arc to the initial state.

10807 Class Project (2016).

2.2 Improvements

2.2.1 Kneser–Ney Smoothing

We improve the bi-gram model by Kneser–Ney Smoothing [1]:

$$p_{KN}(w_i|w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - \delta, 0)}{\sum_{w'} c(w_{i-1}, w')} + \alpha p(w_i), \tag{2}$$

where δ is a constant between 0 and 1. It can decrease the affect of bi-grams which appear very few times. In our work, δ is set to be 0.9.

2.2.2 IBM 2 Model

Although IBM Model 1 has yielded acceptable results, it only uses a very simplified view of the world, where each word in the sentence is translated independently. So we improved this model by implementing the IBM Model 2 that considers word order.

IBM Model 2 is based on the intuition that the reordering between sentences F and E essentially has a canonical word order. We believe this trend is true for the German-English language pairs.

We utilized IBM Model 1 to **pretrain** IBM Model 2, especially parameter $\theta_{f,e}$. Before training IBM Model 2, we trained IBM Model 1 on the same dataset for 5 iterations.

3 Evaluations

	Vanilla	+Smooth	+PLC	+Both
IBM 1	18.70	18.70	18.54	18.59
IBM 2	**19.06**	19.03	19.04	18.99

Table 1: BLEU scores on the validation set. *PLC* denotes "Phrase Length Constraint".

	Vanilla	+Smooth	+PLC	+Both
IBM 1	18.06	18.17	18.05	18.12
IBM 2	18.17	**18.27**	18.12	18.21

Table 2: BLEU scores on the test set. *PLC* denotes "Phrase Length Constraint".

We evaluate our system with three variations, *IBM 2*, adding *Smooth*, and adding *PLC* (Phrase Length Constraint). We report the results in Table 1 and 2. The sequence-to-sequence models are trained on "*train.en-de.low.filt.de*" and "*train.en-de.low.filt.en*" files.

In general, under the same condition, IBM 2 outperforms IBM 1. It indicates that having the prior of canonical word order boosts the alignment. Also, we can see that adding the Kneser–Ney Smoothing improves the performance in the test set, but the BLEU scores of adding the phrase length constraint drop. However, all differences except for the IBM models are marginal. In our submission, we report the system using the IBM 2 model with Kneser–Ney Smoothing.

4 Work Division

Hongliang: Phrase extraction and phrase length constraint.

Keyang: IBM 1 model (`train_model1.py`), IBM 2 model (`train_model2.py`) and Kneser-Ney Smoothing (`train-ngram.py`).

Shuxin: Implementation of `create-phrase-fst.py` file and IBM 2 model.

Conclusion

In this assignment, we finish a symbolic translation model for the German-English translation task. After completing a vanilla model with WSFT, we further modify the model with IBM 2 model (with

pre-trained IBM 1 initialization), Kneser-Ney smoothing and phrase length constraint. We find that IBM 2 greatly improves BLEU scores on both test and validation sets and smoothing works well on the test set. Phrase length constraint reduces the total amount of phrases for WSTF, and the BLEU scores slightly drop on both sets.

References

[1] Y. W. Teh. A bayesian interpretation of interpolated kneser-ney. 2006.