ASSIGNMENT 2

Shrimai Prabhumoye
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sprabhum@andrew.cmu.edu

1 Introduction

Machine translation means to translate text or speech from one language to another. In this assignment we translate a given text according to the probability distribution $p(e|f)$ that the text e in the target language is the translation of a text f in the source language. In this assignment, we will do translation of German to English. We will be using data from IWSLT2016 workshop. We are using data tokenized and filtered using the tokenizer from NLTK (.low.filt), and this tokenized data was further lowercased (.low). In addition, a filtered set with sentences with at least one and no more than 50 words was provided (.low.filt) to us. We will be using those files for our training and experiments.

2 Model

For this assignment, we use IBM Model 1 to model $P(F|E)$ and then use phrase based machine translation to extract appropriate phrases. Finally, we use weighted finite state transducers (WFST) to encode the source text and decode the target text. Hence, the different parts of this assignment are language model, alignment model, phrase extraction and scoring, search for the maximum scoring phrase. We use IBM Model 1 for the language and the alignment model. We first start by learning the parameters of the IBM Model 1 using Expectation Maximization algorithm. IBM Model 1 [1] learns the language and alignment model as follows:

$$P(F|E) \quad = \quad \frac{\epsilon}{(|E|+1)^{|F|}} \prod_{j=1}^{|F|} \sum_{a_j=1}^{|E|+1} P(f_j|e_{a_j})$$

where a_j is alignment for word f_j. $|E|$ and $|F|$ are the lengths of target and source sentences respectively.

Once, we have the probabilities $P(F|E)$ and the alignments, we extract phrases and score them. We have followed Algorithm 6 in [2] for this task. For searching the maximum scoring phrase we use WFSTs. For each phrase pair, we first read the source words one at a time and then according to the transition and score of the arc, we output target words one at a time. For the whole phrase, then the total log probability is calculated by adding the log probabilities for each of the scores of the words in the phrases.

2.1 Modifications

2.1.1 IBM Model 2

One modification that we have implemented is IBM Model 2 [1]. IBM Model 2 takes into account the position of the words in the source and target texts and also their length. The alignment model of the IBM Model 1 is replaced by:

$$P(a_j = t|j, |F|, |E|) \quad = \quad c_{t,j,|F|,|E|}/c_{j,|F|,|E|}$$

In IBM Model 2, we initialize the transitional probability with what is learned from the IBM Model 1.

3 Experiments and Results

We have not restricted our vocabulary of German or English. Hence the vocabulary size of German is 83475 and vocabulary size of English is 41590. Some of the other experiments we tried were changing parameters such as number of epochs while training the IBM model 1 and IBM Model 2.

Configurations	Validation BLEU Score	Test BLEU Score
Max Length of Phrases = 3 No Epochs = 10 No Phrases Train = 1124k Type: German-English Name: IBM Model 1	Avg: **17.66** Unigram: 57.4 Bigram: 24.8: Trigram: 11.8 4-gram: 5.8	Avg: **17.68** Unigram:56.7 Bigram: 24.7 Trigram: 11.8 4-gram: 5.9
Max Length of Phrases = 3 No Epochs = 10 No Phrases Train = 1142k Type: German-English Name: IBM Model 2	Avg: 17.48 Unigram: 57.0 Bigram: 25.1: Trigram: 12.2 4-gram: 6.1	Avg: 17.50 Unigram:56.5 Bigram: 25.2 Trigram: 12.4 4-gram: 6.4
Max Length of Phrases = 3 No Epochs = 5 No Phrases Train = 1158k Type: German-English Name: IBM Model 2	Avg: **18.02** Unigram: 57.4 Bigram 25.4: Trigram: 12.5 4-gram: 6.3	Avg: **17.89** Unigram: 56.4 Bigram: 25.0 Trigram: 12.2 4-gram: 6.3
Max Length of Phrases = 3 No Epochs = 15 No Phrases Train = 1089k Type: German-English Name: IBM Model 2	Avg: 16.97 Unigram: 56.7 Bigram: 25.0 Trigram: 12.3 4-gram: 6.2	Avg: 16.73 Unigram: 56.4 Bigram: 24.9 Trigram: 12.2 4-gram: 6.4

4 Discussions

The baseline IBM Model 1 has BLEU Score of 17.66 on validation set and 17.68 on test set. We initialized IBM Model 2 with the IBM Model 1 parameters after 10 epochs and then ran IBM model 2 for 10 more epochs. This did not beat the baseline scores. Hence, to check if IBM Model 2 under fitted or overfitted, we ran IBM Model 2 for 5 and 15 epochs. The IBM Model 2 with 5 epochs gave the best scores. After 5 epochs, the model over fits. Our best model has a BLEU score of 18.02 on validation set and 17.89 on test set.

One thing, I observed is that the number of phrases extracted from the train set seems to be directly proportional to the BLEU score.

Error Analysis: When I was going through the generated hypothesis of the best model for validation set, I noticed that most of them have semantically the same meaning as the gold hypothesis. But this model makes a lot of grammatical errors as compared to the neural models trained in the previous assignment. This model also generates a lot of '<unk>' characters which then nullifies the meaning of the sentence.

Hypothesis: thus should it 's not surprise that we all of music ...
Gold Sentences: so it shouldn't surprise us that we all sing, ..

Hypothesis: and you are this cultural <unk> as a <unk> imagine. <unk> can help when the sum of thoughts and dreams , myths and ideas , of inspirations – and intuition , from the human imagination ever since the beginnings of consciousness has produced by to be defined .
Gold Sentences: And you might think of this cultural web of life as being an ethnosphere, and you might define the ethnosphere as being the sum total of all thoughts and dreams, myths, ideas, inspirations, intuitions brought into being by the human imagination since the dawn of consciousness.

Number of words generated: Another observation is that the average number of words generated by the phrase based models is higher than the average number of words generated by the neural models.

- Number of sentences in the test set: 1565
- Total Number of words generated by Neural model on test set: 31113
- Total Number of words generated by Phrase based model on test set: 31604
- Total Number of words in the gold test set: 32002
- Average Number of words generated by Neural model on test set: 19.88
- Average Number of words generated by Phrase based model on test set: 20.19
- Average Number of words in the test set: 20.44

This is something that I had expected because in the neural models the model has to predict the '<eos>' token based on statistics. This is not the case in the phrase based models where we map all the phrases of the input text to output text.

References

[1] The ibm models and em algorithm. `http://phontron.com/class/mtandseq2seq2017/mt-spring2017.chapter11.pdf`. Accessed: 2017-02-25.

[2] Phrase-based mt. `http://phontron.com/class/mtandseq2seq2017/mt-spring2017.chapter13.pdf`. Accessed: 2017-02-25.