Symbolic Machine Translation

Samridhi Shree Choudhary and Dheeraj Rajagopal.
Department of Computer Science
Carnegie Mellong University
Pittsburgh, PA 15213
sschoudh@cs.cmu.edu
dheeraj@cs.cmu.edu

Abstract

This document presents a symbolic machine translation model for translating between German and English sentences. IBM Model 1 is used for learning the alignments. The phrases are extracted and converted into Weighted Finite State Transducers (WFSTs). The most probable translation to an input sequence is found by finding the maximum scoring path through the WFST.

1 Introduction

In this assignment we build alignments for translations from German (*"source"* language) to English (*"target"* language). The IBM Translation models form the basis of many Statistical Machine Translation (SMT) Models used today. These generative models aim at assigning a conditional probability $p(f|e)$ to all German/English sentence pairs. The parameters of these models are estimated using the Expectation Maximization **(EM)** algorithm. The alignments learnt are used to extract the phrases. We restricted the phrase length for the model in order to have a tractable state space for the WFST. These phrases are converted into a WFST which is used to find the maximum scoring path (translated sentence) for an input sequence (German sentence). As an addition to this baseline model, we used the predictions from the encoder-decoder model with attention from the previous assignment. For each training sentence pair, we build an alignment depending on the probability distribution obtained from the decoder. These alignments are combined with the alignments obtained from IBM Model 1 and passed on to the phrase extraction module. As a result we get more alignments and therefore more phrases.

2 Pipeline

The symbolic translation models have four components.

- Language Model
- Alignment Model
- Phrase Extraction
- Search for Maximum Scoring Translation

2.1 Language Model

The language model used in the current implementation is an interpolated bigram model. It estimates the probabilities of n-grams using Maximum Likelihood Estimation. The n-grams extracted are further converted into a WFST. This n-gram-WFST is composed later on with the phrase-WFST to

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

find the final translations with maximum score. The maximum scoring English translation for an input German sentence is found by using the following equation:

$$\widehat{E} = \underset{E}{argmax}[P(F|E)P(E)]$$

2.2 IBM Model 1

This probabilistic model learns the alignment probability for the sentence pairs (f, e). The probability is given by:

$$P(f, a|e) = \prod_i P(a_i = j|i, |e|, |f|)P(f_i|e_j)$$

The assumption we make for this model is that all the alignment positions are equally likely. The 'Null' alignment is handled in this implementation by assigning a distortion parameter of 0.2 as a prior for for null alignment position (a_0) in this case. The remaining value of 0.8 is split evenly among the rest of the alignment positions and is set to $\frac{0.8}{N}$.
The parameters of the model are learned by iterative EM updates. A brief description of the 'Expectation' and 'Maximization' steps are as follows:

- Initialize $p(f|e)$ uniformly for each e and f
- E - Step : for all sentence pairs **(f,e)**
  - for all words f in **f** calculate sTotal(f) for each english word e

$$sTotal(f) \leftarrow sTotal(f) + t(f|e)$$

  - Similarly update the counts ($count(f|e)$) and the normalization totals ($total(e)$) for each french-english word pair as:

$$count(f|e) \leftarrow count(f|e) + \frac{p(f|e)}{sTotal(f)}$$

$$total(e) \leftarrow total(e) + \frac{p(f|e)}{sTotal(f)}$$

- M-Step: Update the translation probabilities
  - for all words f in **f** calculate $p(f|e)$ for each english word e

$$p(f|e) \leftarrow \frac{count(f|e)}{total(e)}$$

Given a test sentence, this model aligns the German and English words with the highest translation probabilities calculated above. Training with explicit 'Null' alignments gave us a lower BLEU score and thus we removed it from the final model.

2.3 Phrase Extraction

The phrases are extracted for each sentence pair in a way that is consistent with the alignments found above. The algorithm followed is from the class notes for phrase extraction. Words that are not aligned to any word in the counterpart language are also included by concatenating them to other phrases. This allows phrase-based models to generate these words in many-to-one translations. We update the counts of each of the phrases and the phrase-pairs extracted. These counts are used to calculate the phrase translation probabilities as follows:

$$P(f|e) = \frac{count(f, e)}{\sum_{f'} count(f', e)}$$

The phrases extracted, along with their scores, are written to a text file. This file is used to create a phrase WFST in the next step below.

2.4 Search through WFST

The phrases extracted are converted into a WFST. The procedure followed for this conversion is from the class notes. For each phrase pair, we create a path through WFST as foolows:

1. Read each of the source words
2. Print the target words one at a time
3. Add the log probability from the final state to the initial state (phrase translation probability computed above)

This WFST outputs the maximum scoring translations for the test and validation sets.

3 Experiments and Results

3.1 Baseline

Our baseline model with bigram language model with the IBM Model 1 limited to phrase length 3 running for 10 iterations gave a validation BLEU of **17.66** and a test BLEU of **17.68**.

3.2 Modifications

3.2.1 Iterations of the EM

One of the parameters we experimented with was the number of iterations in the EM. Specifically, we tried iteration=10, iteration=15 and iteration=20 with various settings to obtain the BLEU scores. Our experiments show that BLEU at iteration=15 performs better.

3.2.2 Maximum Phrase Length

We performed experiments by varying the maximum phrase length for aligned phrases. Specifically, we tried phrase lengths of 3 words and 4 words. From our experiments, we found that phrase length of 4 words give the best results at iteration 15 for EM.

3.3 NULL Alignments

One more variation we tried is to examine the effect of having 'NULL' alignments in the translation. We found that 'NULL' alignments did not improve the performance. Rather, we get a decrease in the BLEU score (**16.23**).

3.3.1 Alignments from Neural Model

In our extension to the Symbolic MT system, we combine the alignments from our previous Neural Machine Translation output [1].

For the alignments from neural model, we first trained the neural sequence-to-sequence model with attention. During the decode phase, for each of the input sequence, we use the softmax vector to find $\underset{w}{argmax} P(w_t|w_{s_1 \ldots s_t})$ where w_t is a word from the target sentence. Additionally, we can define a hyperparameter k which considers 'k' top words from the target sentence as alignment. We limit k to 1 for all our experiments.

We consider the obtained target word (k=1) as the corresponding alignment. This serves as an additional alignment file in addition to the alignments we obtain from the IBM Model 1. The hypothesis is the Neural Alignment would give us additional phrases that aren't captured by the IBM Model 1. From our experiments, we observe that the number of phrases increase a lot and hence the computational time to generate valid and test translations also increase a lot. The results for our combined model is shown in table 3.3.1.

[1] We used the model with a Encoder-Decoder model with attention with 15.33 BLEU

RESULTS					
Model Alignment	Iterations	Max Phrase Length	NULL Alignment	Val Score	Test Score
IBM Model 1	10	3	NO	17.66	17.68
IBM Model 1	10	4	NO	17.64	17.72
IBM Model 1	15	3	NO	17.62	17.62
IBM Model 1	15	4	NO	17.66	17.76
IBM Model 1	20	3	NO	17.57	17.6
IBM Model 1	20	4	NO	17.62	17.62
Neural + IBM	10	3	NO	12.43	-

The experiments showed that model that uses alignments from both neural model and the IBM Model 1 underperformed severely compared to the IBM Model 1 baseline. We indeed observed a negative result. The combined model seems to overfit the training data, resulting in a better training accuracy but lower validation and test accuracy. From qualitatively looking at the alignment outputs, we observed that common tokens like '.', '.', 'the' were mapped to several phrases, resulting in a much larger WFST and hence overfitting.

Our best results we obtained from the model with the following hyperparameters

1. Number of EM iterations = 15
2. Phrase length = 4
3. NULL alignments = NO
4. Phrase Alignments from Neural Model = NOT INCLUDED

4 Contribution

Samridhi Shree Choudhary: IBM Model1, Phrase Extraction, Phrase-to-WFST, Experiments(Iterations, Max Phrase Length, NULL alignment), Final Report
Dheeraj Rajagopal : IBM Model1, Experiments(Iterations, Max Phrase Length, NULL alignment), Neural Alignment Expansion, Final Report

References

[1] `http://mt-class.org/jhu/slides/lecture-ibm-model1.pdf`

[2] `http://www.cs.columbia.edu/ mcollins/courses/nlp2011/notes/ibm12.pdf`