HW2: Statistical Machine Translation

Jacob Buckman
`jacobbuc@andrew.cmu.edu`

1 Overview

For this assignment, I implemented a statistical machine translation system, using IBM Model 1 (Wolk and Marasek [2015]) for alignment, phrase extraction (Koehn et al. [2003]), and a WFST to do the translation (Dreyer et al. [2008]). When trained on the IWSLT German-English parallel corpus data, the test-set BLEU score was 14.68.

2 Core Model

This model operates in several distinct phases, which work together to produce a translated version of a given sentence. The model uses a noisy-channel approach to calculate a probablilty for each target sentence E given the source sentence F. This has two components - we need to calculate P(E) and P(F|E). (We can ignore the denominator P(F), which is the same accross all translations.)

To calculate P(E), we use a simple 2-gram language model. For each word, we calculate the unigram probability, as well as the bigram probability going from there to any other word. We store this as a WFST, where each word is transduced to itself, and the weight of the edge is the negative log probability of the unigram or bigram.

Calculating P(F|E) is more complex. First, we need to calculate an alignment between the source sentence and the target sentence. We can do this using the approach in IBM Model 1 - an E/M algorithm for incrementally computing alignments. First, we assign all alignments a uniform probability. We then iterate over the dataset, calculating the total probability mass assigned to each F word, normalized by each E word, as well as the total probability mass doled out by each E word. By summing this up for each F word and normalizing by the total probability mass it received, we can calculate new alignments. After iterating this procedure several times, it eventually converges to an appropriate set of alignments.

The next step was to compute the phrase alignments. As is standard, words in phrases are considered to be aligned if they obey the property of being 'quasi-consecutive' - all words on both sides are either consecutive and aligned to each other, or are aligned to nothing. In order to prevent there being too many alignments such that the computation is infeasible, phrases were limited to 5 words long on both the source and target sides. For each F/E phrase pair discovered, I counted both the pair and the target phrase which it was aligned to; by dividing these two values, we are able to calculate P($F|E$), where F and E are phrase pairs.

In order to discover the most optimal parameterization under these parameters, all phrases were converted into a WFST. This was done by maintaining a single root node, and for each unique phrase, creating a list of cost-0 edges that transduce each source-phrase word to epsilon, followed by cost-0 edges that transduce epsilon to each target-phrase word, and then an edge that returns to the root with a cost equal to the negative log probability of P($F|E$).

Finally, the P(E) n-gram WFST and the P(F|E) phrasal WFST are concatenated. This gives us a WFST for the entire expression P(E)P(F|E). Finally, in order to generate a translation for a sentence, we additionally concatenate a transducer which simply transduces each word in the sentence at 0 cost. When we traverse the minimum-cost set of edges is in the WFST with the Viterbi algorithm, we get the optimal translation.

3 Experimental Design

In my experiments, the model was given sentences in German, and was trained to output sentences in English. The only dataset I utilized was the tokenized, lower-cased IWSLT data. The standard training, validation, and test sets were used. When calculating the IBM Model 1 score, there were 8 iterations of the E/M step. As mentioned earlier, when calculating phrase alignments, any phrases with more that 5 words were not included in the WFST, for speed reasons. Other than this, there were no parameters in the algorithm.

To handle the FST processing, I used Python bindings for the OpenFST toolkit. To evaluate BLEU, I used the MOSES toolkit's multi-bleu.perl script.

4 Results

The validation-set BLEU was 15.36. The test-set BLEU score was 14.68.

References

Markus Dreyer, Jason R Smith, and Jason Eisner. Latent-variable modeling of string transductions with finite-state methods. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1080–1089. Association for Computational Linguistics, 2008.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.

Krzysztof Wolk and Krzysztof Marasek. Real-time statistical speech translation. *CoRR*, abs/1509.09090, 2015. URL `http://arxiv.org/abs/1509.09090`.