Machine Translation
Final Project Interim Report

Jeremy Silver

Thurs. 4/21/15

For this assignment, I used a log-linear model based on four different models trained on the data:

- $P(\text{word}_i \,|\, \text{lemma}_i)$, unigrams.
- $P(\text{word}_i \,|\, \text{lemma}_i, \text{lemma}_{i-1})$, lemma bigrams.
- $P(\text{word}_i \,|\, \text{lemma}_i, \text{word}_{i-1})$, word bigrams.
- $P(\text{word}_i \,|\, \text{lemma}_i, \text{POS}_i)$, unigrams conditioned on part of speech.

I trained each of these models on the data by taking MLE estimates derived from the frequencies in the training data. For simplicity, I used additive smoothing to handle lemmas or bigrams that don't appear in the training data. This smoothing uses a parameter $\alpha > 0$ such that the empirical probabilities are $\hat{P}_i = \frac{x_i + \alpha}{N + \alpha d}$, where x_i is the empirical frequency of n-gram i, N is the total number of observations, and d is the number of distinct word possibilities available given the input. This set of possibilities is the union of the sets of observed words for each model, and it is guaranteed to have at least one member because we assign the lemma itself as a guaranteed output of the unigram model whenever the lemma has never been seen before in training data.

Then for each lemma that appears in the test data, I assign it the following score:

$$\begin{aligned} \textit{score} = \quad & w_1 \log(\hat{P}(\text{word}_i \,|\, \text{lemma}_i)) + w_2 \log(\hat{P}(\text{word}_i \,|\, \text{lemma}_i, \text{lemma}_{i-1})) + \\ & w_3 \log(\hat{P}(\text{word}_i \,|\, \text{lemma}_i, \text{word}_{i-1})) + w_4 \log(\hat{P}(\text{word}_i \,|\, \text{lemma}_i, \text{POS}_i)), \end{aligned}$$

where w_i are weights corresponding to each of the four models. Finally, I output the word choice with the highest score.

I tuned the w_i by simply exhausting over several values and choosing the best setting. Afterwards I tuned α similarly. I found that the bigram models all boosted the results significantly, going from 0.57 (unigram model only) to about 0.65 with any of them. The added utility of having three different bigram models was minor, but still helpful. My optimal settings came out to be $\alpha = 0.5$, $\vec{w} = (0, 0.383, 0.307, 0.307)$, meaning that the unigram was discarded in favor of the stronger models. With this setting I attained a test score of 0.69 on the test data.