Machine Translation — HW 3

Jeremy Silver

Thurs. 3/26

For the first part of this assignment, I implemented the METEOR metric described on the webpage:

$$\ell(h, e) = \frac{P(h, e) \cdot R(h, e)}{(1 - \alpha)R(h, e) + \alpha P(h, e)} ,$$

where h and e are two translations, and P and R are precision and recall.

Since a ternary decision must be made comparing the two sentences to the reference translation, the straightforward way to make this decision is to use the following comparison function for the two machine translations h_1 and h_2, with reference translation e:

$$f(h_1, h_2, e) = \begin{cases} 1, & \ell(h_1, e) > \ell(h_2, e) \\ 0, & \ell(h_1, e) = \ell(h_2, e) \\ -1, & \ell(h_1, e) < \ell(h_2, e) \end{cases}$$

In practice, this decision function never yielded predictions of 0, because even scores that were close were not exactly the same. So I could never effectively match the cases where the human made a decision of 0. To remedy this, I tried adjusting the decision function to the following:

$$f(h_1, h_2, e) = \begin{cases} 1, & \ell(h_1, e) > \ell(h_2, e) + t \\ 0, & |\ell(h_1, e) - \ell(h_2, e)| < t \\ -1, & \ell(h_1, e) < \ell(h_2, e) - t \end{cases}$$

Here t is some small threshold value. I ran the evaluation over several values of t, and unfortunately I found that using any positive value of t led to worse performance. Although I matched some of the human decisions of 0, there were also many false 0's. This makes sense because there is no reason to expect the human's indifference criteria should be particularly related to the METEOR metric.

As for the other parameter α, I tuned it by simply exhausting over the interval $[0, 1]$ using some small increment. I found that $\alpha = .76$ gave me the best comparison with the human judgments.

For my experimental endeavor, I used NLTK's WordNet interface to implement an alternative sentence similarity metric based on lexical synonymy. The fundamental concept of this is the *synset*, which is a sort of lexical "unit", or an equivalence class under the synonym relation. Here is a step-by-step overview of how I implemented my metric:

1. First we choose a synset similarity metric. There are various choices, but the one I used by default was the path similarity, which is a 0-1 normalized score based on the shortest-path distance in the hypernym-hyponym taxonomy.

2. Starting with sentences h and e, we remove from each sentence any words that do not have synsets associated with them. In WordNet, only "content" words (nouns, verbs, adjectives, and some adverbs) have synsets. "Function" words (conjunctions, prepositions, determiners) do not. So we strip away all the non-content words.

3. Since we don't know the true word alignment between the translations, we measure, for each content word w_1 in the shorter of the two sentences, what the greatest similarity is with any word w_2 in the other sentence, and we assign this similarity to w_1. Each word might have multiple senses due to homonymy, so there are multiple synsets associated with them. This creates an inner loop; namely, if w_1 is associated with synsets $s_{1,1}, \ldots, s_{1,m}$ and w_2 is associated with synsets $s_{2,1}, \ldots, s_{2,n}$, the similarity between w_1 and w_2 is the maximum similarity between any $s_{1,i}$ and any $s_{2,j}$.

4. Once each word in the shorter sentence is associated with a similarity score, we average these together in some fashion to get the overall sentence similarity score. I tried both mean and median.

I used this method to generate scores for each sentence pair (h_1, e) and (h_2, e) in the data set. Unfortunately this took a lot of computation time (many hours on my MacBook Pro) due to the quadratic complexity in both sentence length and in number of synsets per word. And in the end, using this similarity metric did not outperform the METEOR metric (I got a score of .484, where METEOR was giving .524).

However, I decided to combine the two metrics linearly with a weight parameter w, so that if $\ell(h, e)$ is the METEOR metric and $\ell'(h, e)$ is the WordNet metric, the combined metric is $(1 - w)\ell(h, e) + w\ell'(h, e)$. I tuned the parameter w in the interval $[0, 1]$ and was surprised to find that $w = 0$ did not actually give the optimal result, but instead it was $w = .07$. I only got a tiny improvement, from .524 to .529, but at least all the effort wasn't for nothing! This demonstrates how combining scores can improve results.

There are many ways I could have changed the parameters to possibly get better results (e.g. the synset similarity metric, or the method of averaging the word scores in the sentence), but the prohibitive computation times deterred me from trying these. I think that while synonym analysis is an interesting way to think about translation evaluation, it is likely that fluency plays a bigger role in a human's assessment of translation quality. Consequently, I think that adding a language model component would probably improve the results even more.