CIS 526 : Homework 3
Decoding

March 6, 2015

1 MODEL IMPLEMENTED

The first method I tried to implement was the METEOR metric. METEOR evaluates translations by computing scores based on word-to-word mapping. It tries to align the hypothesis and the reference sentences by creating a word alignment mapping between them such that every word in a sentence maps to at most one word in the other sentence.
There are 3 types of matches to be made:
1. Exact match
2. Mapping stemmed words (after using the porter stemmer)
3. Mapping words to its synonyms (using WordNet)

I used the porter stemmer and WordNet from the NLTK package to do the mappings. First I got the exact word matches and used a mapping vector to keep track of all the words that have been mapped. Then I stemmed the words in the hypothesis and the reference sentences, mapped the previously unmatched words and updated the mapping vector with the new mappings made. Next, I found all the synonyms for every word in the hypothesis and reference sentences. Then for every remaining unmapped word in the hypothesis sentence, if its synonym occurred in the reference sentence, I updated the mapping vector to indicate a match for that word had been found.

Using the number of matched words, we calculate the precision and recall for every hypothesis-reference pair. The formula is as follows:

$$\textbf{Precision} = \frac{\textbf{Number of matched words}}{\textbf{Number of words in the hypothesis sentence}}$$

$$\textbf{Recall} = \frac{\textbf{Number of matched words}}{\textbf{Number of words in the reference sentence}}$$

Next, we use the values of precision and recall and calculate a harmonic mean using the parameter α.

$$F_{mean} = \frac{\textbf{Precision x Recall}}{\alpha\textbf{Precision} + \textbf{(1-}\alpha\textbf{)Recall}}$$

After mapping all the possible matching words using the above 3 modules, we check if they are in the same relative word order as in the reference sentence, and then introduce a penalty based on that. To do so, we find the minimum number of chunks of matched words in the same word order as the reference sentence.
Using this number of chunks and the number of matches found previously, we can calculate the penalty as follows:

$$\textbf{frag} = \frac{\textbf{number of chunks}}{\textbf{number of matches}}$$

$$\textbf{Penalty} = \gamma.frag^{\beta}$$

The final score for each pair of hypothesis and reference sentence can be calculated using the formula:
score = (1-Penalty) . F_{mean}

Higher score means a better translation. Using the parameters of α, β, γ as 0.9, 3.0 and 0.5 gave me an accuracy of **0.5123**.

Next I tried to take into account the fluency of the hypothesis sentences. I took every possible bigram and trigram pairs (using NLTK) of the two hypothesis sentences and tried to check if they existed in the reference sentence. This gave me a bigram and trigram match count. Then I calculated the recall using these counts and the total number of bigrams and trigrams in the reference sentence. I took a weighted average of the METEOR score and the recall calculated using bigrams and trigrams, and gave more weight to the METEOR score. This definitely improved my accuracy to **0.5372**.
I also attempted to implement a **skip-gram model** which gave me more bigrams and trigrams by skipping over k words. This however did not significantly improve my accuracy and gave me an almost equivalent score.

I tried to implement the LEPOR metric using the paper by Han et al., (2012). The design of the LEPOR metric has 3 main components - a length penalty, n-gram position difference penalty and harmonic mean of precision and recall. I was not able to implement the n-gram position difference properly, hence it did not perform as well as I expected. However I used the idea of the length penalty mentioned in this paper and added it to my METEOR score. Thus my final formula to calculate the score for every hypothesis-reference sentence pair was as follows:

score = (0.6*(1-Penalty)* F_{mean} ***Length Penalty)+(0.4*(bigram recall + trigram recall))**

Using the paramenters of α, β, γ as 0.85, 2.0 and 0.21 gave me an accuracy of **0.561818**.

2 RESULTS SUMMARY

MODEL	α	β	γ	Accuracy
METEOR	0.9	3.0	0.5	0.5123
METEOR + Bigram and Trigram Recall	0.82	1.0	0.21	0.5372
METEOR + Bigram and Trigram Recall (Using 4-skip gram)	0.82	1.0	0.21	0.5371
METEOR + length penalty + Bigram and Trigram Recall	0.85	2.0	0.21	0.561818