1 Introduction

The algorithm that I implemented is a modified version of the METEOR baseline algorithm. I incrementally updated my algorithm by adding different features to it to continuously improve my score. I experimented with METEOR, LEPOR, and ROSE features in order to get to the current score.

1.1 METEOR

Harmonic Mean:

The first step in improving the default code was to implement the simple harmonic mean as described on the homework page. It was a simple yet a rather significant improvement over the default score.

Chunking Penalty:

The next incremental update was to implement the chunking penalty. The chunking penalty is introduced in METEOR to check to which order are the matched word pairs in the correct word order.

This penalty is calculated as: $Pen = \gamma frag^{\beta}$
where $frag = chunks/matches$

Stemming and Synonyms:

In order to improve the chunking count so as to be penalized less, I improved upon the matches in the hypotheses and the reference sentences by incorporating first stem matches between the two sentences and then matching synonyms of the remaining words.

This helps in increasing the number of matches between the two sentences. It also doesn't penalize the hypothesis sentence for using a different word which doesn't change the meaning of the message.

α, β, and γ parameters:

According to the research paper on METEOR, there were a few different values of the parameters that helped them improve the score. I experimented with those values and found the following combination to be the best for my results:

$\alpha = 0.81$
$\beta = 1.0$
$\gamma = 0.21$

1.2 ROSE Features

ROSE is a sentence based, trained metric machine translation evaluation system that uses simple features. I implemented a few of its features and combined them with the METEOR features.

Bigram and Trigram Recall:

Where BLEU is lacking, METEOR and ROSE make up. The primary intuition behind implementing bigram and trigram matching was to check how well formed the hypothesis is as compared to the reference sentence.

The big advantage here being, BLUE cannot implement recall, while I primarily used recall between the two sentences to check the fluency of the hypothesis.

After summing the two recall values, I needed to combine this with the METEOR score already computed. I weighted the two features and summed them together to get the score of the two hypothesis sentences. After some experimentation, I used the following to achieve my score:

score = 0.65 * METEOR + 0.35 * (bigram_recall + trigram_recall)

Failed Features:

1. Bigram and Trigram Precision
2. 4-gram Precision and Recall
3. F-Measure
4. n-gram POS Precision and Recall - Took too long to compute, no significant improvement over the result

1.3 LEPOR

LEPOR focuses on combining two modified factor (sentence length penalty, n-gram position difference penalty) and two classic methodologies (precision and recall). It provides a simple formula to compute its score:

LEPOR = LP * NPosPenal * Harmonic(α Recall, β Precision)

Of the whole LEPOR score, I implemented only the LEPOR length penalty.

The length penalty is give as:

$$\text{LP} = \begin{cases} e^{1-\frac{r}{c}} \text{ if c} < \text{r} \\ 1 \text{ if c} = \text{r} \\ e^{1-\frac{c}{r}} \text{ if c} > \text{r} \end{cases}$$

I combined the LEPOR length penalty with the METEOR score by multiplying the two values.

The final equation I used to calculate the score of the hypothesis sentence:

score = 0.65 * METEOR * LP + 0.35 * (bigram_recall + trigram_recall)