1. Introduction

The goal of this project is to design a metric for automatic evaluation of the translated outputs of two systems and experiment parameters to improve the accuracy of automatic evaluation. The evaluation is sentence level. To this end, we implemented the METEOR metric, length penalty, <more methods we used in the project>

1. Motivation

The first method we chose to implement was the METEOR metric. METEOR evaluates translations by computing scores based on word-to-word mapping. It creates a word alignment mapping to align the hypothesis sentence and the reference sentence. There are 3 types of word matches are applied: exact words match, stemmed words match, and word with its synonyms match. Stemming and synonyms help in increasing the number of matches between the two sentences, since they avoid penalizing the hypothesis sentence for using a different word, which doesn’t change the meaning of the message, from the reference sentence. Further, we implemented chunking penalty as incremental update. Chunking penalty checks if a hypothesis sentence has the same relative word order as in the reference sentence and then introduce a penalty based on that.

1. Approach

The algorithm consists of following steps:

* Find explicit unigram matches between the hypotheses sentence, which from machine translation, and the reference sentence, we used a mapping vector to keep track of all the words that have been mapped. Three types of word matches were found.

1). Exact match:

Only those words that exactly the same in the two sentences were tagged as matched and recorded in the mapping vector.

2). Stemmed match:

The words that were not exactly matched in the hypotheses sentence and reference sentence were stemmed for stemmed matching. We used the “Porter Stemmer” from the NLTK library for the purpose of stemming the word. After mapping the stemmed previously unmatched words, the mapping vector was updated.

3). Synonym match:

The words that were left out after exact match and stemmed watch were considered for synonym match. We used WordNet “synset” library from the NLTK package to find all the synonyms for every unmatched word in the reference sentence. Then for every remaining unmatched word in the hypotheses sentence, we mapped it to the reference sentence if it is a synonym of a reference sentence word, and updated the mapping vector.

* The number of unigram matches found in the hypotheses sentence (h) and reference sentences (e) pair was then used to compute the unigram precision P(h, e) and unigram recall R(h, e). The formula is as follows:

P(h, e) = Number of matched words / Number of words in the hypothesis sentence

R(h, e) = Number of matched words / Number of words in the reference sentence

Next, we used the values of precision and recall with a parameter alpha to compute a harmonic mean (F\_mean).

F\_mean = P(h, e) \* R(h, e)/ alpha \* P(h, e) + (1 – alpha) \* R(h, e)

After mapping all possible matching words, we found the minimum number of chunks of matched words in the same word order as the reference sentence. The number of chunks (c) and the number of unigram matches (m) then used to compute the chunk penalty using following formula:

Penalty = gamma \* ( c / m) ^ beta

Finally, the METEOR score was calculated as follows:

Score = (1 – Penalty) \* F\_mean

1. Data

In the METEOR metric, we used the given data file “hyp1-hyp2-ref” as input file to calculate scores of each hypothesis sentence, besides that, we also downloaded an external database “wordnet” from NLTK library in order to look up the synonyms of words.

1. Code

In this project, we referred Porter Stemmer and WordNet form the NLTK library to do the mapping.

The Porter Stemmer algorithm is a process for removing the commoner morphological and inflexional ending from words in English [1]. Its source code can reached from this link: http://www.nltk.org/\_modules/nltk/stem/porter.html#PorterStemmer

WordNet is a large lexical database for the English language, which was created by Princeton. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms [2]. The source code can be downloaded from WordNet homepage (http://wordnet.princeton.edu/).

1. Experimental Setup

In the METEOR metric, we experimented different value for tunable parameters alpha, beta and gamma to improve accuracy.

1. Results

The table shows how adding each metric to the model improved the accuracy.

|  |  |  |
| --- | --- | --- |
| Metric | Accuracy | Tunable parameters |
| METEOR | 0.523389 | Alpha=0.73 , beta=1.0, gamma=0.21 |
|  |  |  |
|  |  |  |
|  |  |  |

1. Analysis of the Results

The METEOR with 3 types of word match improves the baseline, because it found the words that were used interchangeable in the two sentences, and hence increased the number of matches between the hypotheses sentences and reference sentence.

1. Future Work

In the METEOR, lemmatization is another type of word match that worth to be considered. The goal of both stemming and lemmatization is to reduce inflectional forms and derivationally related forms of a word to a common base form. However, compared to stemming, which usually refers to heuristic process to remove the derivational affixes, lemmatization uses vocabulary and morphological analysis of words to remove inflectional endings only and to return the base or dictionary form of a word[3].

1. Reference:

[1] Porter, Martin. (2006, January). *The Porter Stemming Algorithm*. Retrieved from https://tartarus.org/martin/PorterStemmer/

[2] Princeton University. (2015, March 17). *What is WordNet*. Retrieved from http://wordnet.princeton.edu/

[3] Cambridge University Press. (2009, April 7). Stemming and lemmatization. Retrieved from https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html