CSCI 544 – Natural Language Processing – HW7

Algorithm for Machine Translation Output Evaluation

Improved the accuracy of the machine translation evaluation by implementing a combination of the following functionalities.

1. METEOR – metric M1 **2. WordNet SynSets – metric M2** **3. BLEU – metric M3**

Algorithm

For each example in the dataset of the form hyp1 ||| hyp2 ||| ref

1. Split, Preprocess and Assign

H1 = hyp1 (hypothesis 1)
H2 = hyp2 (hypothesis 2)
REF = ref (reference)
Preprocessing involves tokenization, case converstion and removal of punctuations

2. Compute and Assign Various Metric Scores

/* metric 1 – METEOR scores for each machine translation H1 and H2 wrt Reference REF */
M1_H1 = METEOR_SCORE(H1, REF)
M1_H2 = METEOR_SCORE(H2, REF)

/* metric 2 – Synonym Set scores for each machine translation H1 and H2 wrt Reference REF */
M2_H1 = SYNSET_SCORE(H1, REF)
M2_H2 = SYNSET_SCORE(H2, REF)

/* metric 3 – BLEU scores for each machine translation H1 and H2 wrt Reference REF */
M3_H1 = BLEU_SCORE(H1, REF)
M3_H2 = BLEU_SCORE(H2, REF)

3. Find summation of all metrics

Total_H1 = M1_H1 + M2_H1 + M3_H1
Total_H2 = M1_H2 + M2_H2 + M3_H2

4. Compare total metrics and write evaluation to eval.out

```
If Total_H1 > Total_H2
        write 1
Else If Total_H1 == Total_H2
        Write 0
Else
        Write -1
```

5. Run check.py to determine if the evaluation written to eval.out is in the required format
6. Run compare-with-human-evaluation.py < eval.out to compute machine translation evaluation accuracy against machine translation evaluations performed by human annotations.

METRIC - 1 : METEOR_SCORE(Hypothesis, Reference)

/* Compute the F1 score of each hypothesis h with respect to the corresponding reference ref

Formula: F1 score = (P(h, ref) * R(h, ref) / ((α * P(h, ref) * ((1-α) * R(h, ref))

Precision P(hyp, ref) is the no of words in hypothesis hyp that are also present in refernce ref divided by the total number of words in the hypothesis hyp. **ie P(h,ref) = | h ∩ ref | / | h |**

And Recall R(hyp, ref) is the no of words in the hypothesis hyp that are also present in reference ref divided by the total number of words in the reference ref. **ie R(h, ref) = | h ∩ ref | / | ref |**

α = 0.9 is the tuning parameter that balances the precision and recall */

1. Compute Precision and Recall and F1 score

P(h, ref) = | h ∩ ref | / | h |
R(h, ref) = | h ∩ ref | / | ref |
F(h, ref) = (P(h, ref) * R(h, ref) / ((α * P(h, ref) * ((1-α) * R(h, ref))

/* value of α = 0.9 is determined experimentally */

2. Compute Penalty

//Chunks = no of words in Reference not in Hypothesis
Chunks = [for word in ref where word not in hyp]
//Unigram_Match = no of words in hypothesis that are present in Reference
Chunks = [for word in hyp where word in ref]
Penalty = 0.5*(Chunks / Unigram_Match)

3. Return METEOR score

Return F * (1-Penalty)

METRIC – 2 : SYNSET_SCORE(H, REF)

1. Compare the lengths of the hypothesis and reference

If | length(hypothesis) – length(reference) | < = 3
    Value = Value + 1

2. Check for matching synonyms present in wordnet

For each word w in the hypothesis
    If the synonym of the word w exists in the reference
        Value = Value + 1

3. Check for matching word stems using nltk's Snowball stemmer

For each word w in the hypothesis
    If the stem of the word w exists in the list of stemmed refernce words
        Value = Value + 1

4. Return Value

METRIC – 3 : BLEU_SCORE (Hypothesis, Reference)

1. Find Length of Reference and Hypothesis

R = Length of Reference, H = Length of Hypothesis, BP = basic penalty

2. Validate Lengths

If R == 0 or H == 0
    then return 0

3. Compute Penalty

If R > H
    then BP = 1
else
    BP = e^(1-R/H)

4. Compute N-grams (N = 1,2,3) for both the Hypothesis and the Reference

Ref_unigrams = [each word in Reference]
Ref_bigrams = [consecutive bigrams in Reference]
Ref_trigrams = [consecutive trigrams in Reference]
Ref_quadgrams = [consecutive quadgrams in Reference]

Hyp_unigrams = [each word in Hypothesis]
Hyp_bigrams = [consecutive bigrams in Hypothesis]
Hyp_trigrams = [consecutive trigrams in Hypothesis]
Hyp_quadgrams = [consecutive quadgrams in Hypothesis]

5. Validate length of N-Grams

If length(Hyp_Bigrams) or length(Hyp_Trigrams) or length(Hyp_QuadGrams) == 0
    Return 0

6. Find unigram, bigram and trigram precisions:

UniGram_Prec = | Hyp_unigrams ∩ Ref_unigrams | / | Hyp_unigrams |
BiGram_Prec = 1 +| Hyp_bigrams ∩ Ref_bigrams | / | Hyp_bigrams | + 1
TriGram_Prec = 1 + | Hyp_trigrams ∩ Ref_trigrams | / | Hyp_trigrams | + 1
QuadGram_Prec = 1 + | Hyp_quadgrams ∩ Ref_quadgrams | / | Hyp_quadgrams | + 1
/* 1 is added for smoothing */

7. Compute and Return Geometric mean of the above precisions

Mean = (UniGram_Prec * BiGram_Prec * TriGram_Prec * QuadGram_Prec) ^ (1/3)
Mean = Mean * BP
return Mean

Motivation

BLEU(Bilingual Evaluation Understudy) uses a simple comparison of words in the sense of an n-grams model. This metric helps compare the reference to the hypothesis using the precison of various words and n grams phrases in the corresponding words and phrases in the reference. The algorithm uses this metric along with a smoothing of 1 in the numerator and denominator to account for absence of certain higer order n grams. It helps in finding the n gram overlap between the hypothesis and the reference. Used a variant of BLEU for single sentence evaluations.

The METEOR metric uses precision, recall and F1 scores along with a penalty for shorter evaluations. This metric penalizes those translations that are shorter than the reference. This metric compares the hypothesis with a single metric and calculates the balanced F measure and penalizes by a chunk penalty where a chunk refers to the number of uninterpreted words in the reference. This semantic similarity based matching helps achieve the baseline accuracy of nearly 0.5186.

Using the synsets – set of syonyms for each word provided by wordnet helps in machine translation evaluation by comparing each word to semantically similar or equivalent words in the reference thereby improving the overall precision and coverage. The hypernym's graph of wordnet is used to identify the words in the reference that are synonymous with each word in the hypothesis.

These metrics were used to enhance the accuracies of the automatic machine translation evaluation. Tried to train a classifier using the naïve bayes bag of words model and the logistic regression classifier using words and phrases as features and the values {-1,0,1} as labels. However the classifier did not exceed the baseline threshold accuracy. Code present in classifier.py.