CMPT 413/825 Project
Using ROSE metric for Machine Translation Evaluation

Luiz Peres de Oliveira lperesde@sfu.ca

Turash Mosharraf tmosharr@sfu.ca

Jin Jung sjjung@sfu.ca

Justin Lew jylew@sfu.ca

1 Motivation

The motivation of using regression and ranking based optimisation in the machine translation evaluation problem is to improve the accuracy of the evaluation beyond the baseline implementation. The baseline implementation used the METEOR metric for finding precision and recall (Lavie and Agarwal, 2007).

2 Approach

The approach taken to improve from the baseline is partially based on the implementation of the ROSE metric (Song and Cohn, 2011). The algorithm uses a 27 length feature vector. The feature vector consist of the parameters, preceded by it's ID:

- 1-4: n-gram precision, n=1...4
- 5-8: n-gram recall, n=1...4
- 9-12: n-gram f-measure, n=1...4
- 13: average n-gram precision for sentence
- 14: n-gram score at the document level
- 15-18: n-gram precision for sentence excluding stopwords, n=1...4
- 19-22: n-gram recall for sentence excluding stopwords, n=1...4
- 23-26: n-gram f-measure for sentence excluding stopwords, n=1...4
- 27: average n-gram precision for sentence excluding stopwords, n=1...4

One way to compare the two translations is using n-gram precision and n-gram recall. n-gram precision is the ratio of the count of n-grams in the candidate translation sentence that is in the reference sentence to the counts of all n-grams in the candidate sentence. The n-gram precision is defined as

$$P_n = \frac{\sum_{ngram \in \vec{c}} Count(ngram)[ngram \in \vec{r}]}{\sum_{ngram \in \vec{c}} Count(ngram)}$$

where $\vec{r}$, is defined as the (human) reference sentence, and $\vec{c}$, is defined as the (hypothesis) candidate sentence. Only 1,2,3,4-gram counts are used in the implementation of the algorithm.

3 Data

The data file to train the evaluation model is from hyp1-hyp2-ref. The file consists of a triple (hyp1, hyp2, and ref) where hyp1 and hyp2 are two translations to which is evaluated by the algorithm, along with a reference sentence of the hypothesis curated by a human translator.

The data file dev.answers contains the preference between the two hypothesis translations by a human translator. The numbers correspond to outputs of the function,

$$f(h_1, h_2, e) = \begin{cases} 1, & \text{if } h_1 \text{ is preferred to } h_2 \\ 0, & \text{if } h_1 \text{ is equally good/bad to } h_2 \\ \text{-1}, & \text{if } h_2 \text{ is preferred to } h_1 \end{cases}$$

where h_1 and h_2 are the two hypothesis translation and e is the reference translation.

4 Code

4.1 Pseudocode of modified ROSE metric evaluation

```
Data: $(hyp1, hyp2, ref)$ in $\mathcal{D}$
Result: output $\alpha(h_1, h_2, e)$
Load n-gram model $ngram\_dict$ document level;
Initialize weight vectors $wt$ randomly;
for $(h1, h2, e)$ in $\mathcal{D}$ do
    Preprocess triple ($h_1, h_2$, e);
    Create feature vectors $vc_1, vc_2$;
    Score feature vectors $vc_1, vc_2$;
    $\alpha = \sum_{i=1}^{27} wt_i(vc_{1_i} - vc_{2_i})$ ;
    output $\alpha$;
end
```

4.2 Miscellaneous algorithms used

5 Experimental Setup

Our experiment compares the accuracy between ROSE and the METEOR metric as shown in the baseline. Method 1 was the baseline implementation of the METRO metric. Methods 2 through 8 are modifications to the ROSE implementation (Song and Cohn, 2011) and the modifications to the sentence structure of the data set. Table 1 shows the methods implemented to improve the accuracy of the evaluator.

Method	Description
1	METEOR
2	ROSE, only one feature vector with 13 elements
3	ROSE, added second feature vector with sentences without stopwords
4	Removed all characters with punctuation
5	Included scores for n-grams at sentence level. First feature vector contains 14 elements.
6	Removed all unicode characters and used WordNet to lemmatize sentence input.
7	Used WordNet to check similarities among words
8	Used levenshtein distance

5.1 Results

Method	Time Execution [1]	Dev Score	Test Score
1	1min 33sec	0.510169	0.529
2	11sec	0.512868	0.529
3	50sec	0.517365	0.533
4	48sec	0.519008	0.539
5	55sec	0.520103	0.541
6	4min 51sec	0.523115	0.546
7	5min 43sec	0.526166	0.547
8	6min 17sec	0.530742	0.548

5.2 Analysis of the Results

6 Future Work

Future work that extends the partial implementation of the ROSE metric would be to use a special kernel for categorizing text documents (Lodhi et al., 2002). In order to improve upon ROSE and BLEU, methods for combining scores from partial syntactic dependency matches along with n-gram matches using a statistical parser as presented in the paper (Kahn et al., 2009).

References

Jeremy G Kahn, Matthew Snover, and Mari Ostendorf. 2009. Expected dependency pair match: predicting translation quality with expected syntactic structure. *Machine Translation*, 23(2):169–179.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

Xingyi Song and Trevor Cohn. 2011. Regression and ranking based optimisation for sentence level machine translation evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 123–129. Association for Computational Linguistics.

[1] Time of execution is based on the performance of a 2.2GHz quad-core Intel Core i7 processor MacBook Pro