Reranker

Yating Jing

April 9, 2015

1 Summary

For the reranking task, I tried adding new features to the simple reranker which choose the best translations according to their weighted sum of feature scores. I also tried training a multi-class classifier, but the results aren't very satisfying. The simple reranker yields the best result, which is 0.289.

2 Simple Reranker

2.1 Feature Design

This reranker chooses the translation with the highest weighted sum of the features for each source sentence. Each candidate translation $e \in E(f)$ of an input sentence f has an associated feature vector:

$$h(e, f) = [-\log p_{LM}(e), -\log p_{TM}(f \mid e), -\log p_{TM_{Lex}}(f \mid e),\ bp,\ c(word),\ c(untranslated)]$$

where the added features are:

1. bp: brevity penalty feature
2. $c(word)$: word count feature
3. $c(untranslated)$: untranslated word count feature

More specifically, the brevity penalty feature are defined as:

$$BP = \min\{\frac{|f|}{|e|},\ 1\}$$

where f denotes the source sentence and e denotes the candidate translation. Here assume that translations that are shorter than the source sentence might lead to the loss of information.

$c(count)$ is the number of words in the candidate translation, $c(untranslated)$ is the untranslated portion of the sentence, which is the number of untranslated words divided by the total number of words in the source sentence.

$$\begin{aligned} c(count) &= |e| \\ c(untranslated) &= \frac{c(untranslated\_word\_count)}{|e|} \end{aligned}$$

The total score is defined as the weighted sum of the 6 features above:

$$\begin{aligned} score = \mathbf{w} * h(e, f) &= \sum_{i=1}^{6} w_i f_i \\ &= -w_1 \log p_{LM}(e) - w_2 \log p_{TM}(f \mid e) - w_3 \log p_{TM_{Lex}}(f \mid e) \\ &\quad + w_4 \cdot bp + w_5 \cdot c(word) + w_6 \cdot c(untranslated) \end{aligned} \tag{1}$$

2.2 Result

The feature weights for equation (1) above and the final result are showed in the table below.

Feature Weights						Bleu Score
$w_1(LM)$	$w_2(TM)$	$w_3(TM_{Lex})$	$w_4(bp)$	$w_5(word_count)$	$w_6(untranslated)$	
1.0	0.65	0.8	0.05	0.65	0.6	0.289

Table 1: Feature Weights

3 Training a classifier

3.1 Basic ideas

Data The **training data** is all the candidate translations of the **train.100best** plus the first 50% of **dev+test.100best** along with their corresponding source sentences and reference translations.

Classes The idea is to train a classifier. There are 100 classes in total, since for each source sentence, there are 100 candidates. That is, for each source sentence, each candidate translation belongs to an individual class. The class number indicates which candidate translation is the best. I tired **OneVsRest classifier** and **Logistic regression classifier**.

Feature Vectors For each source sentence instance, the feature vectors are:

$$h[\mathbf{e}, f] = [h_1(e_1, f), h_2(e_2, f), ..., h_{100}(e_{100}, f)]$$

$$h_i(e_i, f) = [-\log p_{LM}(e_i), -\log p_{TM}(f \mid e_i), -\log p_{TM_{Lex}}(f \mid e_i),\ bp,\ c(word),\ c(untranslated)]$$

3.2 Labels

The labels are created based on the similarity of the translation candidates and reference, here different metrics of **word matching** are adopted: METEOR metric, BLEU metric, F1-measure of N-gram matches. For each candidate translation, choose the one that is most similar to the reference translation using the metrics above, and the index($[0, 99]$) of most best translation candidate would be the label for that particular instance.

3.3 Results

The experiment result is unsatisfying, possibly because the number of classes are too large and the training data is not sufficient. The training data matrix is very ”fat and short”, namely, the feature vector for each instance is very long (600 features), but there are only 800 training sentences in total.

Metrics	Bleu Score	
	OneVsRest Classifier	Logistic Regression
METEOR	0.256 ($\alpha = 0.1$)	0.262 ($\alpha = 0.05$)
BLEU	0.263 (bigram)	0.267 (unigram)
F1-measure	0.267 (trigram)	0.268 (unigram)

Table 2: Model performance using different similarity measures

References

[1] Papineni, Kishore, et al. ”BLEU: a method for automatic evaluation of machine translation.” *Proceedings of the 40th annual meeting on association for computational linguistics.* Association for Computational Linguistics, 2002.