Automatic Evaluator

Yating Jing

April 9, 2015

1 Summary

For the automatic evaluator, I tried METEOR metric, BLEU metric, learning various classifiers from training data. The linear regression classifier using feature function 2 yields the best evaluation accuracy, 0.530390.

2 METEOR metric

Use the METEOR metric to compute the harmonic mean of precision and recall, and use the mean to compare two given candidates. The similarity between a candidate hypothesis h and reference *ref* is:

$$l(h, ref) = \frac{P(h, ref) \cdot R(h, ref)}{(1-\alpha)R(h, ref) + \alpha P(h, ref)}$$

where P is the precision and R is the recall. Set $\alpha = \mathbf{0.15}$, the evaluation accuracy is **0.504341**.

3 BLEU

Model This model compares n-grams of the candidates with the n-grams of the reference and the count the number of the matches. The comparison is conducted using **modified n-gram precision**, where a reference word is considered **exhausted** after a matching candidate word is identified. The model consists of brevity penalty BP and a combination of modified n-gram precisions.

Let p_n be the modified n-gram precisions, using n-grams up to length N and positive weights w_n summing to 1. Let c be the length of the candidate translation and r be the length of reference translation.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

Then, the logarithm BLEU score of the given sentence is

$$\log \text{BLEU} = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n log\, p_n$$

where w_n is set to be $\frac{\mathbf{N-n+1}}{\sum_{\mathbf{1}}^{\mathbf{N}}}$, where lower order grams are assigned higher weights, which yields a better result than the uniform weights. Additionally, rather than use the BLEU score directly, I use it to compute the final accuracy under the **METEOR** metric before comparison.

Results The evaluation accuracy is even lower than the accuracy under simple METEOR metric when $N > 1$, which showed in the table below. This might be due to the fact that BLEU is more suitable for corpus evaluation rather than single sentence evaluation. Also, this is an unsmoothed model, a smoothed one is claimed to be able to yield better results.

N	α	BLEU accuracy
1 (unigram)	0.65	0.509426
2	0.5	0.499100
3	0.8	0.476377
4	0.8	0.456273

Table 1: BLEU accuracy (under METEOR metric)

4 Classifiers

I tried training different kinds of classifiers using python machine learning library. I used supervised training and the labels are set to be all the references available.

4.1 Feature Functions

The basic idea is to project to input onto a feature space and then pipe the feature matrix into the classifier. The features contains: **string matching features** and **word count features**. More specifically, string matching features consist of **modified n-gram precision, modified n-gram recall and modified F1-measure**, the meaning of *modified* is the same as in section 3.

I designed two different feature functions to experiment with.

Feature Function 1 In this setting, string matching features consist of **weighted sums** of modified n-gram precision, modified n-gram recall and modified F1-measures if different n. Thus for each candidate translation hypothesis there are **four** features.

Let P_n be the modified n-gram precisions R_n be the modified n-gram recalls and F_n be the be the modified n-gram F1-measures, using n-grams up to length N and positive weights w_n summing to 1. Let c be the candidate translation and r be the reference translation. The equations for the three string matching features are:

$$P_N = w_n P_n = w_n \frac{|h \cap r|}{|h|}$$

$$R_N = w_n R_n = w_n \frac{|h \cap r|}{|r|}$$

$$F_N = w_n F_n = 2 w_n \frac{P_n \cdot R_n}{P_n + R_n}$$

Here w_n are set to be **uniform** weights.

The word count feature here is set to be **brevity penalty** since a candidate would be considered less preferable if its length is shorter than the reference.

$$BP = \min\{\frac{|r|}{|c|},\ 1\}$$

Feature Function 2 In this setting, each feature vector contains **3N + 1** features, which are n-gram precision, n-gram recall, F1-measure (where $n = 1, 2, ..., N$) and the word count feature(brevity penalty feature). The feature vector can be denoted as :

$$\vec{f} = [P_1, R_1, F_1, P_2, R_2, F_2, ..., P_N, R_N, F_N, BP]$$

4.2 Binary Classifiers

The rationale of using binary classifiers is that it is very unlikely that two candidate translations have exactly the same translation level. So in this method output 0 is omitted.

Linear Regression The results trained by a linear regression classifier are showed in the table below. Since feature function 2 obviously yields better results than function 1 as N increases, not all the results are given in the table.

N	Accuracy	
	Feature function 1	Feature function 2
1 (unigram)	0.523936	0.523936
2	0.526400	0.526752
3	0.528160	0.529568
4	0.527613	0.529451
5	0.527926	0.529568
6	0.526557	0.529607
7	/	0.529725
8	/	**0.530390**
9	/	0.530272

Table 2: Linear regression classifier accuracy

Logistic Regression The results trained by a logistic regression classifier are showed in the table below.

N	Accuracy	
	Feature function 1	Feature function 2
1 (unigram)	0.525618	0.525618
2	0.526244	0.525540
3	0.526205	0.525735
4	0.525383	0.525814
5	0.523858	0.525305
6	0.526557	0.526518

Table 3: Logistic regression classifier accuracy

Support vector machine The results trained by a support vector machine classifier are showed in the table below.

N	Accuracy	
	Feature function 1	Feature function 2
1 (unigram)	0.525149	0.525149
2	0.525110	0.526009
3	0.525462	0.526087
4	0.524406	0.526635
5	0.523349	0.527456
6	0.522645	0.526087

Table 4: SVM classifier accuracy

4.3 Multiclass Classifier

I used the **OneVsRestClassifier** from sklearn to train the classifier, which can be used to predict non-binary labels. This model fits one classifier per class, and for each classifier the class is fitted against all the other classes. Here I tried feature function 2 only. The results are:

N	Accuracy (Feature function 2)
1 (unigram)	0.524210
2	0.527143
3	0.527261
4	0.527065
5	0.526791
6	0.527613

Table 5: OneVsRestClassifier accuracy

References

[1] Philipp Koehn. Statistical machine translation. 2009. Cambridge University Press.

[2] 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.

[3] Song, Xingyi, and Trevor Cohn. "Regression and Ranking based Optimisation for Sentence Level Machine Translation Evaluation." *Proceedings of the Sixth Workshop on Statistical Machine Translation.* Association for Computational Linguistics, 2011.