Machine Translation
Rerank

Haitang Hu
hthu@cs.jhu.edu

April 8, 2015

1 Rerank Methods

1.1 Minimum Bayes-Risk Decoding

Minimum Bayes-Risk Decoding[1] aims to minimize the expected loss of translation errors. By applying specific metrics(**BLEU**, **WER**) as loss function, we could form our MBR decoder to improve reranking performance on Machine Translation System.
The minimum Bayes-Risk can be expressed in terms of *loss function* and our decoder generated results $\delta(F)$.

$$R(\delta(F)) = E_{P(E,A,F)}[L((E,A),\delta(F))]$$

Where the expectation is taken under the distribution of $P(E,A,F)$, which means the true joint distribution.
For this problem, we have a well known solution if given the loss function and a distribution

$$\delta(F) = \arg\min_{E',A'} \sum_{E,A} L((E,A),(E',A'))P(E,A|F)$$

where E is the translation sentence, A is a alignment under the translation (E,F). But this ideal model is far from reality, since we don't have the true distribution for our $P(E,A|F)$. Here, we compromise to our statiscal methods, to guess the distribution through $N-best$ list we have, and now the model becomes

$$\hat{i} = \arg\min_{i\in\{1,2,\ldots,N\}} \sum_{j=1}^{N} L((E_j,A_j),(E_i,A_i))P(E_j,A_j|F)$$

where $P(E_j,A_j|F)$ can be represented as

$$P(E_j,A_j|F) = \frac{P(E_j,A_j,F)}{\sum_j^N P(E_j,A_j,F)}$$

Note that $P(E_j,A_j,F)$ now is just a empirical distribution under given $N-best$ list.
This model suggests that we should look into all our $N-best$ list, and select the *average* one as our results, since the *average* one always gives us less surprise, or *risk*.
Also, it might be worth citing the paper's proof, that if we use a indicator function on our loss function, then MBR reduced to the MAP estimator.

$$\delta_{MAP}(F) = \arg\max_{(E',A')} P(E',A'|F)$$

This is intuitive, since MAP just use point estimate which assumes all our distribution density peaks at the point. Instead, MBR gives a more smoothed distribution.

1.2 Feature Extension

It is natural that we should not only depends on our translation model, language model and lexical model score. Here we encode another 2 belief into our features, to get a better representation of our domain knowledge. First, we consider the word counts, an intuitive way to encode our belief is to penalize with respect to the difference of length $\delta(c, r)$ between candidate and reference. The second feature is simply the number of untranslated Russian words, notated as $u(c)$. So, we have our model score to be following

$$s(c) = \lambda_{l(c)} l(c) + \lambda_{t(c)} t(c) + \lambda_{lex(c)} lex(c) - \lambda_{\delta(c,r)} \delta(c, r) - \lambda_{u(c)} u(c)$$

Here we have 5 parameters, and we should choose them to fit best to our training data.

2 Implementation

2.1 Metric

Generally, **BLEU** will be used as our loss function. Since **BLEU** score always lies in the range of $(0, 1)$, so we could encode our loss function to be

$$L((E_j, A_j), (E_i, A_i)) = 1 - BLEU((E_j, A_j), (E_i, A_i))$$

Recall, we also need to specify our posterior distribution. Here we specify it to be

$$P(E_j, A_j|F) = \log(l(E_j)) + \log(t(E_j|F)) + \log(lex(E_j, A_j|F))$$

Also, there is another point worth mentioning, that is the **BLEU** can both applied on *string level* and *word level.* We will show the performance comparison later.

2.2 Efficiency

Since for each $N - best$ list, we need to at least loop N^2 times, since we have a pairwise loss, so it is neccesary to implement it in a smarter way. Here we employ a matrix method to avoid to loop twice for computing normalize constant and pariwise loss.

3 Evaluation

3.1 Result

Method	Score
baseline	0.2735
baseline($lm = -1.0, tm = -0.65, lex = -1.3$)	0.2817
feature ext($lm = -1.0, tm = -0.65, lex = -1.3, c = 1.03, u = 0.1$)	0.2893
MBR	0.2916
MBR + word count	**0.2918**

Table 1: Result
c, u stands for word count weight and untranslated words weight

3.2 Evaluation and Optimization

As we can see here, simply tuning the parameter of *baseline* system gives us a big improvement, which shows the huge gain we could get from *MERT* or *PRO*(But I did not encode them).
Next, we look at he feature extension, which again raise a lot score, since we actually encode the significant reason to express our belief, which happens to be right.
We gain the best score by using MBR method with counting *word count* feature into our posterior distribution, this shows the we could benifit from more feature under MBR setting. Put it another way, this suggests combining *MERT* could benifit the MBR method.

References

[1] Shankar Kumar and William Byrne. *Minimum Bayes-Risk Decoding for Statistical Machine Translation*, 2011.