Chinese-English Statistical Machine Translation System
CMPT 825 Course Project "Team Infinity"

Chandan Mishra
cmishra@sfu.ca

Pranav Dixit
pranavd@sfu.ca

Neel Tiwari
neelt@sfu.ca

Abstract

This paper is for Chinese to English statistical machine translation system with emphasis on feedback exchange between decoder and reranker. We have used future cost based beam search decoder and PRO reranking algorithm with ordinal regression to build our decoder-reranker system. Our idea here is to use significant features such as language model, translation model and alignment score for our decoder and use feedback loop from the reranker system to improve translation quality. We have improved on the PRO algorithm by combining ordinal regression for better learning in the reranking phase of the system. Our experiment shows +3.5 BLEU score improvement as compared to baseline system having only decoder. We have used Moses tokenizer which further improved the BLEU score by +2.0.

1 Motivation

Feedback exchange is widely used for many machine learning tasks for enhancing the performance of the system. As we are using a perceptron-like algorithm to learn weights, the idea of using the approach and see it work, interested us. The feedback exchange module here, however limits the reranker system to local features, features used by the decoder, but it helps in learning the weights for the decoder system, resulting in better translation.

In addition, having already worked on both the modules of our decoder-reranker system, gave us the base to build on.

2 Approach

We are using a phrase-based statistical machine translation method for translation of source Chinese sentence f to target English sentence e.

$$\begin{aligned} e &= \arg\max_{e} P(\mathbf{e}|\mathbf{f}) \\ &= \arg\max_{e} P(\mathbf{e}) * P(\mathbf{f}|\mathbf{e}) \end{aligned}$$

As, the number of candidate target sentences for a source sentence can be exponentially large, we use a future cost based beam search to narrow down on the potential best translation of the source sentence. The beam search decoder generates n-best target candidates for each source sentence. For our model, we use language model feature, 4 translation model feature, distortion feature and IBM Model 1 alignment feature.

The decoder generates n-best target candidate sentences, which are feed into the reranker system to learn the weights that favors the candidates which are similar to the 4 references for the source Chinese sentence. We use the BLEU metric to find similarity between the candidate target sentences and 4 reference sentences. The reranker system learn the weights for the features using a perceptron-like algorithm PRO Algorithm (Hopkins and May, 2011). We have enhanced the learning of the weights via PRO Algorithm using ordinal regression (Shen, 2004). The learned weights are used by the decoder as the feedback from the reranker system to generate new n-best target candidates for the reranker system. We iterate the process till we get a minimal improvement in the BLEU score on the dev data. The final

weights generated by the decoder-reranker system are used by the decoder on the test data to generate better translation of source Chinese sentences to target English sentences.

The figure below describes the flow of our approach:

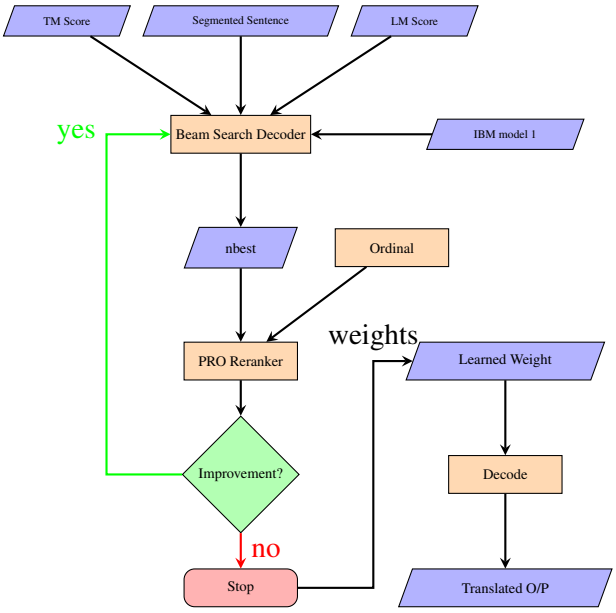

Figure 1: Basic flow Chart of our system.

2.1 Beam Search Decoder

The phrase-based machine translation is a NP-hard problem as it has exponential number of choices to choose from to find the best translation of the source sentence. Beam Search (Collins, 2013) is an optimization algorithm to reduce the search space of a problem. We implement the beam search using a data structure, hypothesis, which holds the coverage of the source sentence words, the features for the hypothesis, the score of the translation and the future cost. Depending on the number of source sentence words covered, we have stacks of hypothesis with the *s* best future costs. For each hypothesis, we choose the phrases that can be added to the hypothesis to create new hypothesis. We add the new hypothesis to a stack only if we do not have a similar hypothesis with a better score in the stack. Once, we cover all the source sentence words, we have a stack with hypothesis for the candidate target language sentences. We generate the n-best candidates from the n-best hypothesis from the last stack, stack representing that all the source sentence words have been covered.

2.2 Future cost

Using Top-N or beam decoder, better translations may appear to be poor translations at the beginning of stack decoding, and they are often pruned out. To better handle this, Future Cost Estimation (Koe,) was used to measure how expensive it would be to translate the rest of a sentence based on the current hypothesis. Future cost for input sentence can be precomputed using dynamic programming and only looked up while decoding.

$$future\_logprob = future\_cost + local\ logprob$$

2.3 PRO Algorithm with ordinal regression

The PRO Algorithm (Hopkins and May, 2011) is used to rerank the n-best candidate target sentences generated by the the decoder. The PRO algorithm is a perceptron-like algorithm which learn feature weights for the model using the observations made on the pair-wise samples generated from the n-best candidates. The algorithm compares the model score of the candidates in the sample with the BLEU metric of the candidates. It learns the weights so as to make the model score consistent with the BLEU score. We have enhanced our PRO reranking algorithm with ordinal regression. The ordinal regression is based on the fact that some pair of candidates are better for learning than others. Let e_i be the candidate ranked at the i^{th} position for the source sentence, where ranking is defined on the quality of the candidates. For example, knowing e_{100} is better than e_{101} may be less useful than to know that e_1 is better than e_{100}. Hence, we only consider pair of candidates in the sample for learning if their ranks are considerably far away from each other.For our model, we set the updating condition as $y_{i,j} * 2 < y_{i,l}$, and $y_{i,j} + 8 < y_{i,l}$ or $y_{i,j} + 20 < y_{i,l}$ depending on the stack size used for our decoder, where $y_{i,j}$ represents the rank of the j^{th} translation of the i^{th} source sentence. We have tried other methods too, for better learning of weights such as PRO algorithm with splitting. It was due to the better results we got with ordinal regression during our reranker assignment, we follow the ordinal regression approach for our model.

Algorithm 1 Get Sample of PRO Algorithm with ordinal condition

```
0: procedure GETSAMPLE(NBEST, TAU, ALPHA)
0:   sample ← to return sample
0:   nbest ← shuffle nbest
0:   for tau times do
0:     s1 ← First candidate
0:     s2 ← Second candidate
0:     if ordinalRank(s1) − ordinalRank(s2) > 20 and ordinalRank(s1)/ordinalRank(s2) then
0:       if Bleu(s1) > Bleu(s2) then
0:         sample.append(s1,s2)
0:       else
0:         sample.append(s2,s1)
0:     else
0:       Continue
0:   return sample.
```

2.4 IBM Model 1 Score

We have used IBM model 1 score as one of the features for our decoder-reranker system. To start with, we build the model *Pr(f | e)* using the conditional probability *t(f | e)* where f is the source language and e is the target language. Let the source sentence be **f** = (f_1,...,f_I) and target sentence **e** = (e_1,...,e_J). Every source word f_i has an alignment a_i which corresponds to an target word e_{a_i}. The alignment vector is given by **a** = (a_1.....,a_I).

$$P(\mathbf{f}|\mathbf{a},\mathbf{e}) = \prod_{i=1}^{I} t(f_j|e_{a_i})$$

The IBM model 1 score is the sum of all the alignments for the sentence.i.e

$$P(\mathbf{f}|\mathbf{e}) = \sum_{a} P(\mathbf{f}|\mathbf{a},\mathbf{e})$$

$$= \sum_{a} \prod_{i=1}^{I} t(f_j|e_{a_i})$$

$$= \prod_{i=1}^{I} \sum_{a} t(f_j|e_{a_i})$$

And, for non existing word pair we have used $1/10^8$ as smoothing value.

3 Experiment

3.1 Data

The training data is taken from the following sources:

- Hong Kong Parliament parallel corpus
- GALE Phase-1 Chinese newsgroup data

And, it was further divided in various size for the project, like Toy (2 K sentence pair), Small (20 K sentence pair), Medium(100 K sentence pair) and Large (2.3 M sentence pair). For our experiment, we started with toy data for integrating various module. Our experiment is with first **500 lines** of dev data for tuning weights. And, evaluated performance on full test data set. Language model for training and test is same and filtered from large LM **en.gigaword.3g.filtered.train_dev_test.arpa.gz**

	Dev
TM	/dev-filtered/rules_cnt.final.out
Source	/dev/all.cn-en.cn
Reference	/dev/all.cn-en.en0, en1, en2, en3

Table 1: Dev Data.

	Test
TM	/test-filtered/rules_cnt.final.out
Source	/test/all.cn-en.cn
Reference	/test/all.cn-en.en0, en1, en2, en3

Table 2: Test Data.

For IBM Model 1 score, we trained our model on the source and reference pair available in **Medium data set** having 100 K pair.

3.2 Code

We mainly used codes from homework 3, homework 4 and homework 5. To automate feedback looping mechanism, we written utility shell script and integrated *multi-bleu.perl* for final BLEU score computation. We have provided script to use moses tokenizer as well.

align.py - EM algorithm for lexical word alignment, saved model as pickle to use and get alignment score during decoding time.

	Test
HW3	align.py
HW4	decoder.py, models.py
HW5	learn.py, bleu.py, score-reranker
External	multi-bleu.perl, moses tokenizer

Table 3: Codes from homework.

decoder.py - beam search decoder with Future cost estimation algorithm;
learn.py - PRO reranking algorithm with ordinal regression.
bleu.py - Modified existing file to handle four references.

3.3 Experimental Setup

For our experiment, we have defined four different kind of models as mentioned below, In all of these settings,we are taking stack size (s) = 10, Number of translation per phrases(k) = 6 and distortion limit of 6 or more with penalty. In our reranker we are by default using ordinal regression to better classify positive and negative examples. Also, we provided shell script **./run.sh** for running below models.

- **Baseline:** Decoder having basic six features
  - LMScore: the language model score for tranlation candidate
  - ReorderingScore: the sum of the distortion penalties for this translation
  - p(fle): the inverse phrase translation probability
  - lex(fle): the inverse lexical weighting
  - p(elf): the direct phrase translation probability
  - lex(elf): the direct lexical weighting
- **Baseline + IBM Model 1 Score:** Additionally added IBM model 1 alignment score, as in homework 5 we have observed that it was a significant feature which improved our BLEU score by minimum of +1.2.
- **Baseline + Reranking:** Having six features with feedback loop mechanism, we loop only for 3 epochs considering time taken by decoder.
- **Baseline + IBM Model 1 Score + Reranking:** This is combination of model 2 and 3.

4 Results

Results in Table 4 is calculated using ***multi-bleu.perl*** and also verified by ***score-reranker.py*** Using parameters stack size s = 10, number of translation k = 5 and distortion limit = 6 and trained on top 500 dev data.

Models	**BLEU**
Baseline	6.98
Baseline+IBM Model 1	7.39
Baseline+ Reranker	7.23
Baseline+IBM Model 1+Reranker	9.7

Table 4: Machine Translation Evaluation trained on 500 Dev data

Further on model 4, we done one trial run with stack size 100, k=5 and with and without distortion limit and got results shown in Table 5. It clerly shows without limiting distortion BLUE score improves and it is obvious as Chinese and English alignment is quite distorted. However, one point we want to highlight even though we are not restricting distortion but giving distortion penalty in both cases.

	Baseline+IBM Model 1+ Reranker
d=6	10.35
d=NA	11.75

Table 5: with s=100, k=5, dev data = 500

Sample Translation:

Source Sentence: 国际 原子能 总 署 已 警告 , 假如 平壤 拒绝 放弃 核子 计划 , 它 可能 要求 安理会 采取 行动 。

Baseline translation: pyongyang unscr abandon nuclear warning may request refused international atomic energy scheme , if taken action , it had has .

Baseline+ IBM Model 1+ Rereanker: security council refusal to give up nuclear if it may request an atomic energy total taken action scheme has already warned department , pyongyang , international.

As we can see last translation is having meaningful sentence compared to baseline translation, and reason for that is better language model score in latter.

5 Analysis of the Results

There are few interesting observation we have found during our experiment.
1. **Distortion** - There are wide distortion possible between Chinese and English phrases, as we can observe from given alignment files. So, by allowing distortion with higher value improves BLEU score which can be verified using results in Table 5.

2. **Reranking** - Here, PRO reranking algorithm behaves inconsistently as it is based on random sampling. During HW 5, we observed that PRO reranking was not that efficient. However, with introduction of ordinal regression, where we are assigning rank to candidates and taking two pair which are having rank difference of at least 8(in case of StackSize=8) or 20(if stack size = 20), it helps in better learning of weights.

3. **Stack Size** - Again, larger stack size leads to better translation as we compared in Table 4 and Table 5

4. **Tokenization of output translation** - Using Moses tokenizer, we are able to improve BLEU score by upto +2.0. After applying tokenizer on model 4 output with 9.7, we got improvement of +2.29 to **11.99**.

5. **UNK Chinese words** - We obeserved that there were not many unknown chinese words after translation. Hence, did not went ahead to handle the unknown words as that would have resulted in only

6 Future Work

We would like to handle UNK Chinese words, and made code capable of running on multi processor or GPU. Also, given a superior hardware we can try on larger dataset which will surely improve BLEU score.

References

Michael Collins. 2013. *Phrase-Based Translation Models*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1352–1362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Libin Shen. 2004. Discriminative reranking for machine translation. In *In HLTNAACL 2004*, pages 177–184.