# Adam Group's Experiements on Chinese-English Machine Translation: Final Project of CMPT413 Winter 2016

**Lyken Zhu**
lykenz@sfu.ca

**Jetic Gu**
jeticg@sfu.ca

## Abstract

As the final project, we as Adam Group implemented a machine translation system with two different approaches: one with the classical phrase-based decoder and reranker approach, another one with neural network(Neural Machine Translation). We compared the results of these two different approaches and concluded that the Neural Machine Translation is much more superior(BLEU score 0.218) than the traditional phrase-based approach(BLEU score 0.07)

## 1 Movitation

The implementation starts with our assignments. Our teams implementation of decoder and reranker performed quite well, so our first idea is to make them work together, and also try out some new ideas that we didnt have time for while doing the individual assignments.

We have also attempted to do experiments on a larger scale, by using the large dataset and Gigaword language model, but resulted in failure due to the limitation of our computer. The large dataset combined with Gigaword language model requires too much memory, so we had to abandon the plan. We have also attempted the feedback loop between the decoder and reranker because we wanted to improve the performance.

Last but not least, we also looked at a recent paper (Manning, 2015) which proposed a state-of-the-art model based on seq2seq LSTMs.

## 2 Approach

### 2.1 Phrase-based Approach

Our first approach is to use our original decoder implementation and reranker implementation. We modified the decoder to generate features and a list of n-best sentences while decoding. We used a textbook translation model:

$$e = \underset{e}{argmax}\ Pr(e|f) = argmax \sum_{a} \underset{TM}{Pr}(f, a|e) * \underset{LM}{Pr}(e)$$

The algorithm we used to perform decoding is simple. Each term, we choose a phrase from the source sentence, and add it to every sentence in the stack. After each term, prune the stack according to their language model scores and translation model scores so that we do not get exponential running time. Here is the pseudocode:

```
# Initialise stack1
add emptySentence to stack1
# start computation
for i=1 to length(f) do
 for phrase in all the phrases from
   f do
  for sentence in stack1 do
   if phrase not overlap with
     sentence then
    newSentence =
      combine(sentence, phrase)
    if
      translationComplete(newSentence)
      then
     add newSentence to answerSet
    else
     add newSentence to stack2
 stack1 = prune(stack2)
 stack2 = emptyStack
# Prune to N-Best
prune(answerSet)
```

After retrieving a set of all the answers(answerSet in the algorithm above), we use the features and a weight vector to rerank these sentences, and the sentence with the highest feature score will be the final output.

### 2.1.1 The Features

We used the classic features for our rerankers, including the following:

- $LMScore$: the language model score for this candidate
- $ReorderingScore$: the sum of the distortion penalties for this translation
- $p(f|e)$: the inverse phrase translation probability
- $lex(f|e)$: the inverse lexical weighting
- $p(e|f)$: the direct phrase translation probability
- $lex(e|f)$: the direct lexical weighting

All of the features are generated during decoding. For reordering score, we used the following equation to calculate the score:

$$Score = \sum_{d} log(a^{|d|})$$

where d is the distance of the original position of the two adjacent phrases in the target language. We tried several values and settled down for 0.9.

### 2.1.2 The Score for Reranking

The total score for reranking is calculated by the following formula:

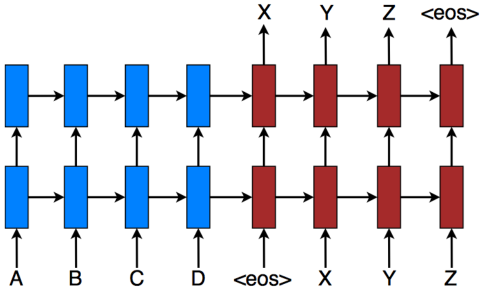
$$Score = Feature^{T} * Weight$$

whereas the weights are calculated by our rerankers learner. The learner implementation is exactly the same with our assignment implementation, which is basically a pairwise ranking optimisation algorithm given in HW5.

## 2.2 Neural Machine Approach

Neural Machine Translation (NMT) is a powerful model which achived several state-of-the-art peformances in large-scale translation tasks such as English-French (Luong 2015) and English-Germen (Jean et al. 2015). After studying the results of these papers, we believe it might also show a competitive performance on Chinese to English translation.

The sequence to sequence model is pretty straight-forward: first the neural machine reads through all of the source words until the $< eos >$ is reached; then RNN units will be used to perform end-to-end trainning.



As illustraed in the figure above, its size only depends on the length of total words. Because the Gigaword phrase-table is to large for our machines, and the fact that after feeding enough data, a large neural network has the ability to generalise the rules well by itself, we did not use the Gigaword phrase-table.

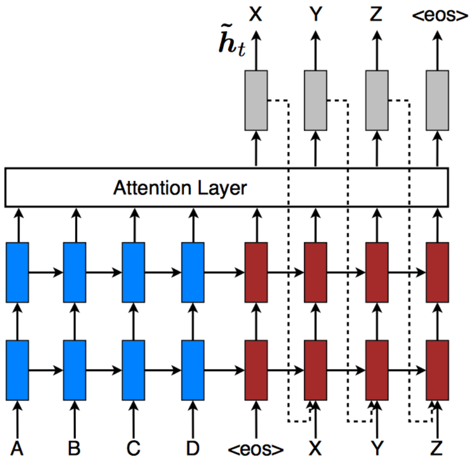
According to the paper (Mnih et al., 2014), the concept of ”attention” improves the score by allowing models to learn alighnments between different modalities. Luong also further explored the use of attention-based architectures of NMT, and achived state-of-the-art in Germen to English translation. Hence, we studied and decided to utilise their design: one global and local attention layer above for a variable length-alignment.

The $\bar{h}_s$ is the source hidden state. and $\bar{h}_t$ is the current targe hiddent state.

$$a_t(s) = align(h_t, \bar{h}_s) = \frac{exp(score(h_t, \bar{h}_s))}{\sum_{s'} exp(score(h_t, \bar{h}_{s'}))}$$

Where

$$score(h_t, \bar{h}_{s'}) = \begin{cases} h_t^T * \bar{h}_{s'} & dot \\ h_t^T * W_a * \bar{h}_{s'} & general \\ h_t^T * tanh(W_a[h_t; \bar{h}_{s'}]) & concat \end{cases} \quad (1)$$



## 3 Data

| item | **Data** |
|---|---:|
| phrase table | filtered-phrase-table |
| language model | nlp-data/lm/en.tiny3g.arpa |
| Input | nlp-data/medium/train.cn |
| Reference | nlp-data/medium/train.en |

Table 1: test-data for phrase-based model

| item | **Data** |
|---|---:|
| alignment | nlp-data/meidum/alignemnt |
| Train | nlp-data/large (last 40k) |
| Valid | nlp-data/medium |

Table 2: test-data for NMT model

## 4 Code

Our code consists of two parts: the original implementation, and the NMT.

The code of our phrase-based approach implementation is under the directory ‘project‘, which are all written by ourselves or were provided by the instructor to use during HW1-5.

Our Neural Machine Translation implementations main programme is under the directory ‘seq2seq‘. It utilises RNN/GRN frameworks from (Graves, 2015) and the skeleton code from Harvard NLP group (Manning, 2015)

## 5 Experimental Setups

### 5.1 Phrase-based approach

As naive Chinese speakers, we’ve noticed some imperfection of the segmented data provided for us, such as ”Lei Shenglong Yiyuan”(Shenglong Lei MP) is segmented as ”Leisheng Long Yiyuan”(Thunder Long MP). So in addition to using the segmented data, we also used re-segmented data using Stanford segmenter(), and did the following

1. Generate N-best using the segmented Chinese sentences.

2. Generate weights using the N-best file using our own decoder.

3. Use the our rerank to compute proper weights

4. Use weights to perform translation

### 5.2 Neural Machine Approach

In this approach, we don’t need phrase table. All sentences are preprocessed to hdf5 file consumable for Torch.

1. Filter those abitarily long sentence (more than 50 words)

2. Map all appeared EN words and CN phrases into a number

3. Feed into Recurrent Neural Network, and wait patiently.

## 6 Results Analysis

| Method | **BLEU** |
|---|---:|
| baseline (using filtered table) | 0.071 |
| baseline (using stanford’s segmenter) | 0.076 |
| neural machine approach | **0.218** |

Table 3: Performance Comparasion

Our design of the phrase-based method consists of a lot of parts(segmentation, phrase-table selection, decoding, reranking), and each part is equally influential and therefore important. During the early tests we have discovered that better segmentation means better quality of translation, but during our implementation of reranker in HW5, we have also noticed irregularities of the reranker on specific datasets, where the error rate might not drop but instead increase as we train the weights. Also, datasets with the same score and close resemblance could also result in dramatically different weights, plus it so happens, weve also learned from other groups that the potential benefit of using feedback loop could be very limited. We suspect that this is caused by the amount and type of features we used, and therefore considered the possibility of using neural network to do the translation.

According the the results above, it is within reason to reach the conclusion that Neural Machine Translation does have its superb qualities. Instead of manually selecting and generating features, the End2End model is fully capable of automatically generating its own feature and conduct learning, which justifies its marvellous performance.

## 7 Future Work

As for phrase-based approach, though we would not expect any huge improvement, there are still lots we could do. For example, one could look into the possibility of using a much more efficient decoder implementation, or the option of generating more features for reranking. But as far as BLEU score are concerned, we believe that the neural machine translation might be a better choice. There are still lots of potential options, like embeling segmention into translation, making the whole pipeline end-to-end, adopting residual connections(He et al., 2015) which is similar to attention-based to neural network to NLP tasks and inversitaging how depth and length affect the performance of neural network in NLP(Pandey and Dukkipati, 2016) .

## References

Alex Graves. 2015. *Generating Sequences With Recurrent Neural Networks*. arXiv:1308.0850, University of Toronto.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. *Deep Residual Learning for Image Recognition*.

Minh-Thang Luong Hieu Pham Christopher D. Manning. 2015. *Effective Approaches to Attention-based Neural Machine Translation*. NIPS, Computer Science Department, Stanford University, Stanford, CA 94305.

Gaurav Pandey and Ambedkar Dukkipati. 2016. *To go deep or wide in learning?* NIPS, Indian Institute of Science, Bangalore 560012, India.