# Chinese to English Machine Translation Specialized in Game of Go and Cooking Recipe with Limited Data

Juexing Wang
UMass Amherst
Amherst, MA
juexingwang@cs.umass.edu

Weiqiu You
UMass Amherst
Amherst, MA
wyou@cs.umass.edu

## Abstract

*In this project, we built an RNN with Attention model for machine translation from Chinese to English on two special tracks: Chinese cooking recipe and Game of Go. To evaluate our translation results in these two tracks, we also proposed a new evaluation criteria "BLEU-POT", where POT stands for "Penalty on Terms", since traditional BLEU score can't reflect the translation accuracy on critical terms. It is a serious error if we translate terms in our domain wrong, but not as serious if we translate a general word slightly different. We also came up with a data augmentation method to solve our problem of limited data. We randomly sampled two or three words from the Go terms list, and concatenated them together to make a large amount of random sentences. We pretrained our model on the augmented dataset and then trained on terms and sentences, and compared the result with training just on terms and sentences. The models pretrained with augmented data avoided overfitting and outperformed models trained only on terms and sentences. Finally, we compared our best model with Google Translate on test set in both BLEU and BLEU-POT scores. Our best model works better than Google Translate in books in Game of Go.*

## 1. Introduction

The problem we are trying to research in is Machine Translation. Many interesting Chinese topics are also popular in the world, such as Go, traditional Chinese medicine, Taoism, and Chinese cuisines. Translating them from Chinese to English can provide better access for people who are interested in it. Take Go as an example, there are abundant resources in Chinese, Japanese, and Korean that are not available for English speakers. Current top general machine translator such as Google Translate doesn't do a good job in translating them to English. We are hoping to provide a tool to help Go players from the world to read Go books in Chinese. We are also going to collect Chinese menus and finetune our models to translate Chinese cuisine names to English. There are a lot of good Chinese cuisines in China, but the western world only know a few, and not very authentic. We are hoping to help non-Chinese speaking people be able to access Chinese cuisines by providing better translation of them to English.

There are two approaches that we implemented to compare. The first approach is [4], using RNN encoder and decoder. The decoder also incorporates attention mechanism to decide which parts of the source sentence to focus on. The second approach we are using is Transformer model in [7], which abandons the RNN architecture and only uses multi-head self-attention. We will first preprocess Chinese data, and then run these two architectures on Chinese-English translations parallel datasets. And then we will evaluate the performance and compare to state-of-the-art. Finally, we will finetune our models on Go and cooking data and compare the translation result with normal Google translation.

For general Chinese-English translation, there are several public available datasets. We choose "News Commentary V13" as our general dataset since it contains a large amount of parallel English and Chinese sentences. The size of the dataset also acceptable for us to work with locally. For our special track: Game of Go and Cooking. We will build these special datasets by combining the online dataset with dataset made by ourselves. Go and cooking terms dictionaries are also on Sensei's Library website. However, size of these special datasets still not enough for us to train a good model and work well in these special track. So we are going to investigate in data augmentation methods to help us get better results.

## 2. Background/Related Work

This work experimented with two models, RNN (with Attention)[4] model and Attention only[7]model. Our RNN model bases on [6], using RNN from encoder to decoder.

See Figure 1.

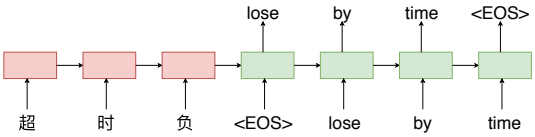


Figure 1. The graphical illustration of the encoder-decoder process of proposed model [6]

We changed the unit of RNN from LSTM to GRU, so that we can improve our computation speed and get similar outcome. GRU use less memory space and iteration with less parameters. Either LSTM or GRU can allow the RNN model [4] learn more information from time-sequence.

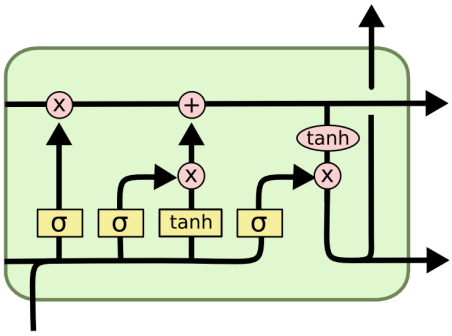


Figure 2. LSTM unit [6]

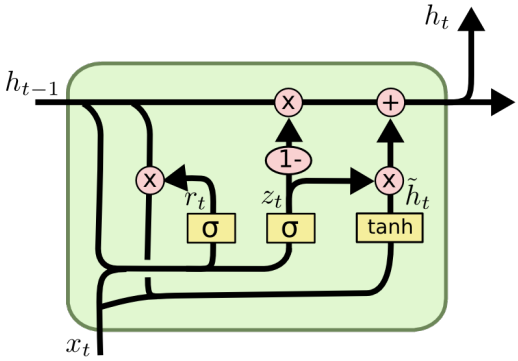


Figure 3. GRU(Gated Recurrent Unit) [6]

Limited data has already become the common issue for neural machine translation in some special tracks, [8] applied some methods that improved the BLEU score significantly. But in our project, BLEU score is not the most relevant criteria to evaluate our result. Sometimes translation result with high BLEU socre still have serious problem. Take Game of Go as examplewe want to translate "黑棋形薄" as black's shape is thin from Chinese to English, but our model tend to transfer it to white's shape is thin. These outcome is close but the meaning is totally opposite, so we decided to design another method to evaluate our outcome with BLEU.

## 3. Approach

### 3.1. Model

We have two models: RNN (with attention) model and Attention only model. But the performance of Attention only model is restricted by the size of our train dataset. So we decided to focus on our RNN model and improved its performance on our two special tracks. Figure 4 shows our RNN model's architecture.

In this model, we choose Chinese characters as input and output will be English words. Input words will be transformed with word embedding, and encoded by GRU units. On another part, our tag <SoS>will go through word embedding, drop out, and then concatenate with original word embedding output and have softmax applied to the output of concatenation. Output of GRU and Softmax will go through BMM (batch matrix-matrix product of matrices) and Output of BMM will pass through all the Attention step. We will choose word with highest probability as the first output word. Generated word will go through the next decoder step to predict next output word. The process keeps going until we get the whole sentence.

### 3.2. Evaluation Methodology

BLEU (BiLingual Evaluation Understudy)[5] score is the universal method of evaluation for machine translation tasks.

$$\text{BLEU} = \min(1, \frac{\text{output-length}}{\text{reference-length}})(\prod_{i=1}^{4} precision_i)^{\frac{1}{4}}$$

We use the BLEU score implemention in the Python nltk package. We use a 0.5 weight on unigram and 0.5 weight on bigram because our sentences and terms are short in general.

In addition, we are proposing a new penalty method POT (Penalty on Terms) in evaluation aside from the BLEU score. Since we are working on special tracks, we especially don't want to get translations of terms in our special

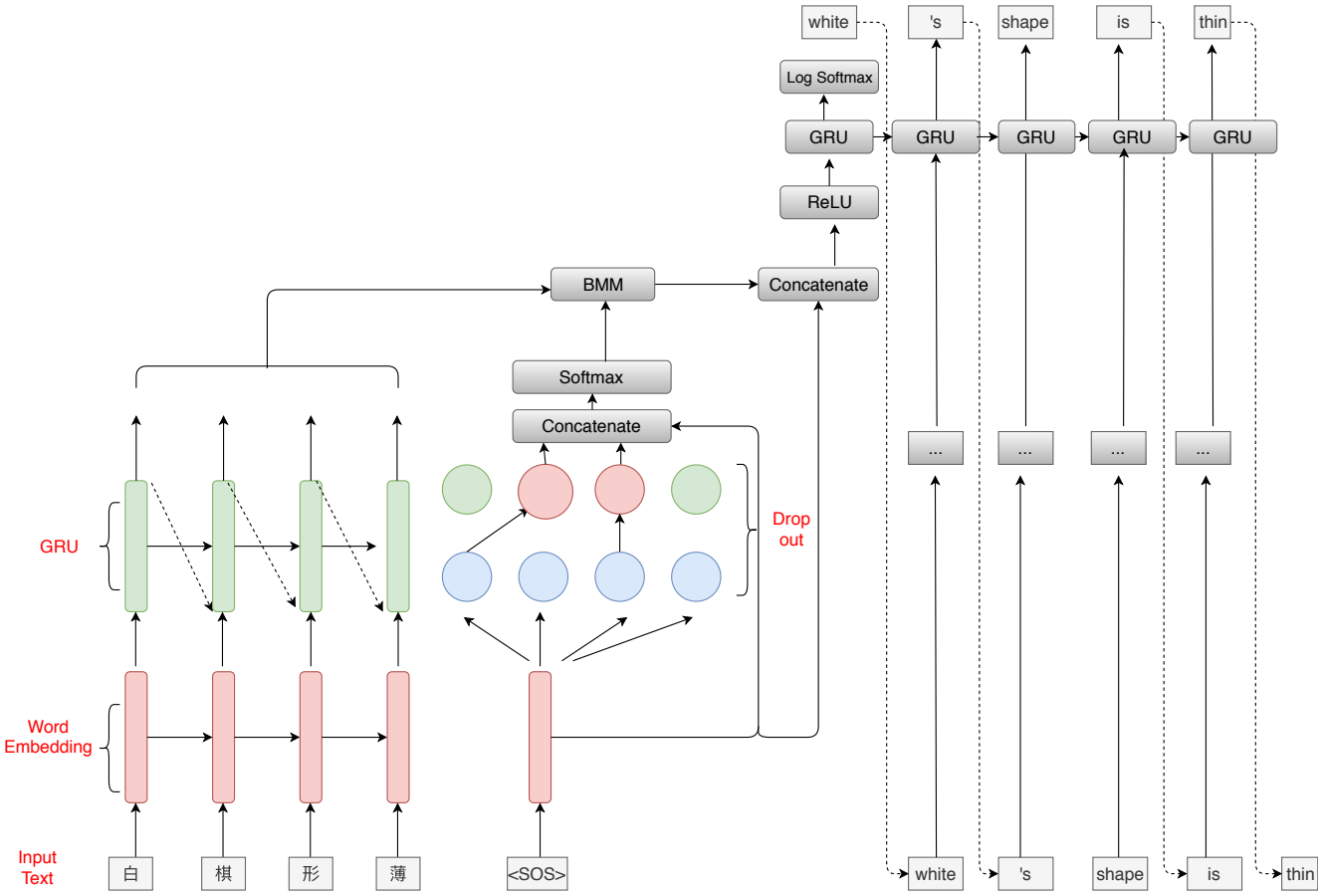


Figure 4. Our model's architecture (Seq2Seq model with GRU and Attention)

tracks wrong. It is not a serious issue if we omit a "the" or say words in different order, but it is a serious error if we translate terms in our domain wrong. As we have a list of terms, we are able to determine if the terms are wrong or omitted. Therefore, we are proposing a new penalty to take away score from translation if the candidate translation does not include terms that appear in the reference translation.

The penalty score is calculated by:

$$\text{reduction}_i = \frac{\text{\# missing word from candidate in term } i}{\text{term}_i\text{-length}}$$

$$\begin{aligned} L &= \text{length of reduction} \\ &= \text{\# terms in reference but not in candidate} \end{aligned}$$

$$\text{POT} = \sum_{i=1}^{L}(\text{redution}_i) \cdot \frac{L}{\text{reference-length}}$$

This POT score will be deducted from BLEU score, and form our final BLEU-POT score for evaluation.

$$\text{BLEU-POT} = \text{BLEU} - \text{POT}$$

## 4. Experiment

### 4.1. Dataset

We first acquired News Commentary V13 from WMT 18[3] as our general dataset. For our special track in game of Go, we obtained terms from Sensei's Library[2], a collaborative web site about and around the game of Go. There are 729 terms in total. We were not able to get parallel translation data for Go books, so we annotated 400 sentences ourselves, assisted by Google Translate. We also added data that contain "black", "white", and different number of stones, which is 30 parallel texts. We also formed a 10 sentence test set for evaluation. For our special track in


Figure 5. Word Cloud of Cooking Recipes

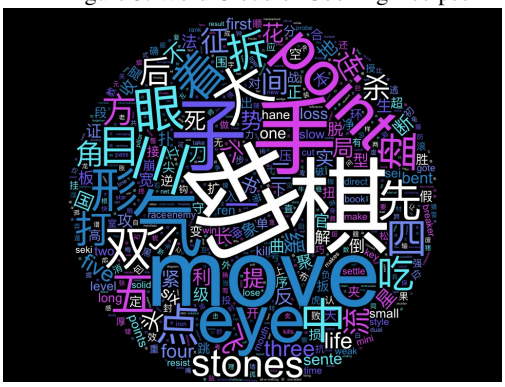


Figure 6. Word Cloud of Game of Go

Chinese cuisine names, we also obtained names of Chinese cuisines from [1]. There are 426 terms in total. As the context for cuisine names is only in the menu, we didn't train to translate sentences that contain Chinese cuisine names, but formed 10 new cuisine names as a test set.

### 4.2. Training

First we experimented with the two different models, and Transformer didn't work well, so we decided to stick to the RNN with Attention model.

The RNN with Attention model trained solely on Go terms or on cuisine names is able to overfit the training set and get 100% accuracy on training set.

To be able to translate sentences in Go books, We first pretrained on the News Commentary dataset and then on the terms, hoping for the model to learn the sentence structures through the general News dataset, and then apply the sentence structures to translate sentences that contain Go terms. However, the result was not satisfying. As there are too many unrelated words in the dataset, the model was not able to converge to translate Go terms correctly. However, training solely on Go terms and the 400 sentences we translated does not generalize well.

We came up with a data augmentation method. We randomly sampled two or three words from the Go terms list, and concatenate them together to make random sentences. This way, the model is able to learn to distinguish meanings of words when they are combined with other words, thus able to generalize to test dataset better. We randomly sampled 10000 "sentences" that contain two terms and 10000 that contain three terms, resulting in a 20000 augmented dataset.

Since we didn't translate sentences that contain Chinese cuisine name, there are three different settings of training without learning the syntactic structure on third training set:

1. Training on combination of all terms, numbers for 20 iterations.
2. Training on combination of all terms, numbers for 50 iterations.
3. Training on randomly sampled terms, numbers for 20 iterations.

We tested four different settings of training on Game of Go:

1. Training on combination of all terms, numbers, and sentences for 20 iterations.
2. Training on combination of all terms, numbers, and sentences for 50 iterations.
3. Training on randomly sampled sentences for 10 iterations, and then training on combination of all terms, numbers, and sentences for 10 iterations.
4. Training on randomly sampled sentences for 10 iterations, and then training on combination of all terms, numbers, and sentences for 10 iterations, and then training on just the 400 sentences for 10 iterations.

### 4.3. Evaluation

We used BLEU score and BLEU-POT score to evaluate on the test set. The results are shown in Table 1and Table 2.

Generally, the models pretrained with 20000 randomly generated fake Go book sentences perform much better than the models trained only on terms and sentences, because the real sentences are too few. Pre-training on the augmented data allows the model to learn to distinguish terms from each other better in the context of a sentence. However, the disadvantage is that it doesn't allow the model to learn better grammar. This problem cannot be solved unless we have more parallel sentences data.

For our cooking recipe, this method seems to be in an awkward position as we can see in Table 1. On the one hand, output of its translation gets higher BLEU and BLUE-POT score than the traditional train method on out test set. On the other hand, it generate more meaningless output. Take 腌鸡 (Salted Chiken) as an example, our model translate it to "salted chicken salted with chicken salted chicken

| | | | BLEU | BLEU-POT |
|---|---|---|---|---|
| 1 | Chinese Original | 烤鱼 | | |
| | English Reference | backed fish | | |
| | All 20 | baked crisp crucian carp | 7.458340731200295e-155 | 7.458340731200295e-155 |
| | All 50 | baked crisp crucian carp | 7.458340731200295e-155 | 7.458340731200295e-155 |
| | Random 20 | broiled fish braised | 8.612150057732663e-155 | 8.612150057732663e-155 |
| 2 | Chinese Original | 酱牛肉 | | |
| | English Reference | braised beef with soy sauce | | |
| | All 20 | braised beef with soy sauce | 1 | 1 |
| | All 50 | braised beef with soy sauce | 1 | 1 |
| | Random 20 | braised beef with soy sauce | 1 | 1 |
| 3 | Chinese Original | 腌鸡 | | |
| | English Reference | salted chicken | | |
| | All 20 | boiled chicken slices | 0 | -0.5 |
| | All 50 | boiled chicken slices with mustard | 6.6709427497276e-155 | -0.5 |
| | Random 20 | salted chicken salted with chicken salted chicken | 0.10482848367219184 | 0.10482848367219184 |
| | | salted corn boiled chicken with chicken salted | | |
| 4 | Chinese Original | 糖醋鸡片 | | |
| | English Reference | fried chicken slices with sweet and sour sauce | | |
| | All 20 | fried sliced whelk with chicken | 6.341195631391024e-155 | 6.341195631391024e-155 |
| | All 50 | fried sliced whelk with chicken liver | 7.55774771366971e-155 | 7.55774771366971e-155 |
| | Random 20 | fried sliced with chicken and sweet chicken sour sauce | 0.3118047822311618 | 0.3118047822311618 |
| 5 | Chinese Original | 清蒸全鸡 | | |
| | English Reference | steamed whole chicken | | |
| | All 20 | steamed chicken | 9.047424648113057e-155 | 9.047424648113057e-155 |
| | All 50 | steamed chicken | 9.047424648113057e-155 | 9.047424648113057e-155 |
| | Random 20 | steamed whole fish boiled chicken with chicken | 0.26726124191242434 | 0.26726124191242434 |
| 6 | Chinese Original | 清炖猪肉 | | |
| | English Reference | braised pork in clean soup | | |
| | All 20 | braised pork | 0.22313016014842982 | 0.22313016014842982 |
| | All 50 | braised pork | 0.22313016014842982 | 0.22313016014842982 |
| | Random 20 | braised venison in clear soup braised braised | 0.099258333397093 | 0.099258333397093 |
| | | braised braised braised braised braised | | |
| | | braised braised braised braised braised | | |
| | | braised braised braised braised braised | | |
| | | braised braised braised braised braised | | |
| | | braised braised braise | | |
| 7 | Chinese Original | 芝麻鱼 | | |
| | English Reference | fried fish with sesame | | |
| | All 20 | fried fish with sesame | 1 | 1 |
| | All 50 | fried fish with sesame | 1 | 1 |
| | Random 20 | fried fish with sesame | 1 | 1 |
| 8 | Chinese Original | 油焖牛肉 | | |
| | English Reference | braised beef | | |
| | All 20 | braised beef | 1 | 1 |
| | All 50 | braised beef | 1 | 1 |
| | Random 20 | braised beef with brown sauce | 0.316227766016838 | 0.316227766016838 |
| 9 | Chinese Original | 炒大虾 | | |
| | English Reference | stir-fried prawns | | |
| | All 20 | stir-fried prawns with fresh mushrooms | 0.316227766016838 | 0.316227766016838 |
| | All 50 | stir-fried prawns with fresh mushrooms | 0.316227766016838 | 0.316227766016838 |
| | Random 20 | stir-fried prawn slices | 8.612150057732663e-155 | 8.612150057732663e-155 |
| 10 | Chinese Original | 软炸鱼肝 | | |
| | English Reference | soft-fried fish liver | | |
| | All 20 | soft-fried pork's liver | 1.0547686614863434e-154 | 1.0547686614863434e-154 |
| | All 50 | soft-fried pork's liver | 1.0547686614863434e-154 | 1.0547686614863434e-154 |
| | Random 20 | soft-fried pig 's liver soft-fried fish fish fish fish | 0.06078306738548309 | 0.06078306738548309 |
| | | fish fish fish fish fish | | |
| | | fish fish fish fish fish | | |
| | | fish fish fish fish fish | | |
| | | fish fish fish fish fish fish | | |

Table 1. Comparisons between Different Models for Cooking Recipe Translations (Highlighted: best score of each sentence)

| | | | BLEU | BLEU-POT |
|---|---|---|---|---|
| 1 | Chinese Original | 开劫是败着 | | |
| | English Reference | creating ko is the losing move | | |
| | All 20 | losing move is the losing move | 0.632455532 | -0.034211135 |
| | All 50 | losing move is the losing move | 0.632455532 | -0.034211135 |
| | Random 10 + All 10 | losing move is the losing move | 0.632455532 | -0.034211135 |
| | Random 10 + All 10 + Book 10 | in this is the losing move | 0.632455532 | -0.034211135 |
| 2 | Chinese Original | 黑棋形薄 | | |
| | English Reference | black ’s shape is thin | | |
| | All 20 | thin shape is thin | 0.550695315 | 0.350695315 |
| | All 50 | black ’s shape is thin | 1 | 1 |
| | Random 10 + All 10 | black ’s shape is thin | 1 | 1 |
| | Random 10 + All 10 + Book 10 | black ’s shape is thin | 1 | 1 |
| 3 | Chinese Original | 保留变化 | | |
| | English Reference | preserve variation | | |
| | All 20 | variation | 0.367879441 | -0.132120559 |
| | All 50 | variation with hane | 0.577350269 | 0.077350269 |
| | Random 10 + All 10 | variation with extra leg | 0.5 | 0 |
| | Random 10 + All 10 + Book 10 | preserve variation | 1 | 1 |
| 4 | Chinese Original | 反而可将黑方三子吃掉 | | |
| | English Reference | on the contrary can capture black ’s three stones | | |
| | All 20 | on the contrary can capture black ’s two stones | 0.816496581 | 0.70538547 |
| | All 50 | on the contrary can capture white ’s two stones | 0.623609564 | 0.17916512 |
| | Random 10 + All 10 | on the contrary can capture black ’s three stones | 1 | 1 |
| | Random 10 + All 10 + Book 10 | on the contrary can capture black ’s three stones | 1 | 1 |
| 5 | Chinese Original | 黑虽强行做活 | | |
| | English Reference | although black can forcefully make a live | | |
| | All 20 | make black to make live | 0.519227675 | 0.519227675 |
| | All 50 | make black to live | 0.409081435 | 0.409081435 |
| | Random 10 + All 10 | black ’s strong group | 0.236183276 | 0.236183276 |
| | Random 10 + All 10 + Book 10 | black is make live | 0.409081435 | 0.409081435 |
| 6 | Chinese Original | 白形有余味 | | |
| | English Reference | white shape has aji | | |
| | All 20 | white has shape is incomplete | 0.774596669 | 0.524596669 |
| | All 50 | white has the aji | 0.866025404 | -1.383974596 |
| | Random 10 + All 10 | white ’s shape is incomplete | 0.632455532 | 0.382455532 |
| | Random 10 + All 10 + Book 10 | white has the aji | 0.866025404 | -1.383974596 |
| 7 | Chinese Original | 这一手很好 | | |
| | English Reference | this move is good | | |
| | All 20 | this move is good | 1 | 1 |
| | All 50 | this move is not good | 0.632455532 | 0.632455532 |
| | Random 10 + All 10 | this move is good | 1 | 1 |
| | Random 10 + All 10 + Book 10 | this move is good | 1 | 1 |
| 8 | Chinese Original | 正解图夹是本手 | | |
| | English Reference | in the solution diagram clamp is the proper move | | |
| | All 20 | in the solution diagram diagonal move is the proper move | 0.730296743 | 0.619185632 |
| | All 50 | in the solution diagram diagonal move is the proper move | 0.730296743 | 0.619185632 |
| | Random 10 + All 10 | in the solution diagram is very move is the proper move is the proper move | 0.478091444 | 0.366980333 |
| | Random 10 + All 10 + Book 10 | in solution diagram move is proper move | 0.401681509 | 0.290570398 |
| 9 | Chinese Original | 黑三子不能被吃 | | |
| | English Reference | black three stones ca n’t be captured | | |
| | All 20 | black can be connect through | 0.423947621 | -4.718909522 |
| | All 50 | black can capture white ’s two stones | 0.534522484 | 0.391665341 |
| | Random 10 + All 10 | black ’s three ca n’t capture black | 0.3086067 | -1.977107586 |
| | Random 10 + All 10 + Book 10 | black ’s three stones n’t live | 0.309091423 | 0.166234281 |
| 10 | Chinese Original | 成为净活 | | |
| | English Reference | become to live unconditionally | | |
| | All 20 | is possible to make live | 0.632455532 | 0.632455532 |
| | All 50 | unconditional life to live | 0.40824829 | 0.40824829 |
| | Random 10 + All 10 | unconditional is unconditionally live | 0.707106781 | 0.707106781 |
| | Random 10 + All 10 + Book 10 | become is unconditionally | 0.585045365 | 0.585045365 |

Table 2. Comparisons between Different Models for Go Book Sentences Translations (Highlighted: best score of each sentence)

| # | Chinese Original | English Reference | Random 10 + All 10 + Book 10 | Google Translate |
|---|---|---|---|---|
| 1 | 开劫是败着 | creating ko is the losing move | in this is the losing move | Open robbery is defeated |
| 2 | 黑棋形薄 | black's shape is thin | black's shape is thin | Black chess shape thin |
| 3 | 保留变化 | preserve variation | preserve variation | Change change |
| 4 | 反而可将黑方三子吃掉 | on the contrary can capture black's three stones | on the contrary can capture black's three stones | Instead, eat Black's three sons. |
| 5 | 黑虽强行做活 | although black can forcefully make a live | black is make live | Black is forced to do life |
| 6 | 白形有余味 | white shape has aji | white has the aji | White has a taste |
| 7 | 这一手很好 | this move is good | this move is good | This hand is very good |
| 8 | 正解图夹是本手 | in the solution diagram clamp is the proper move | in solution diagram move is proper move | Positive solution is the hand |
| 9 | 黑三子不能被吃 | black three stones can't be captured | black's three stones n't live | Black three can't be eaten |
| 10 | 成为净活 | become to live unconditionally | become is unconditionally | Become a net |

Table 3. Comparisons of Best Model with Google Translate (Green is Good, Red is Bad)

salted corn boiled chicken with chicken salted" like Mcdull. Cooking recipes is different with Game of Go in some aspect. It focuses more on combinations of terms, which are shorter than sentences. Increasing length of train set will lead our model to work badly on some test data with short cuisine name. Our model tends to generate long sentence after pretrained on the augmented train set. This phenomenon proves that it is necessary to learn the syntactic structure even though our input data are short terms.

With our limited data, we are already doing pretty good in generalizing sentences from Go books as we can see in Table 2. In Example 4, the training sentence says "capture white's two stones", so training on just terms and sentences for 50 iterations lead to overfit and the model cannot translate "black's three stones" correctly.

Some sentences are not coherent, because our training set is too small that it doesn't cover all the sentence structures.

When translating a long sentence, if all the other words are correct, but the terms involved is wrong, the sentence loses its meaning. In the 9th example, most of the sentence is translated correctly, but the key word "clamp" (a type of move in go) is not translated out. This will make the reader confused and not able to know the right move.

Our new evaluation criteria is still not perfect. We can see from example 6 that "white has the aji" is a better translation than "white has shape is incomplete" with reference "white shape has aji", but it got penalized for missing the word "shape". In this case, the non-penalized BLEU score becomes a better criteria.

We compared our best model with Google Translate on test set. Google Translate can translate more coherent sentences like in Table 3, but the key terms are usually wrong, unless the key terms are everyday words and have the same meaning as the everyday meanings. For instance, in Example 4, "子" means "stone" in game of Go, but can also mean "son" in a general sense. It doesn't make sense to translate the sentence as "eat black's three sons", which sounds terrifying, while it should be translated as "capture black's three stones".

In Figure 7, we can see how our models outperform Google Translate in this specific task of translating Go Books in both BLEU and BLEU-POT scores. We can also

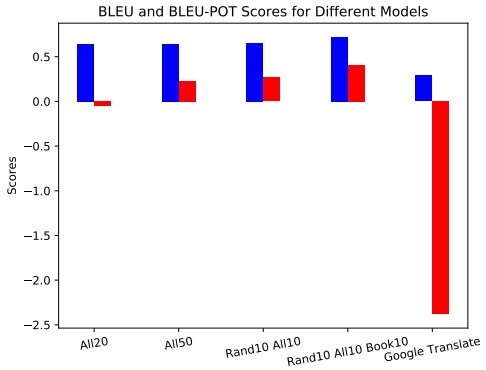


Figure 7. BLEU and BLEU-POT Scores for Different Models for Go Data

see that, the insignificant difference between our four models when evaluated with BLEU score becomes significant when evaluated with BLEU-POT score. BLEU-POT score becomes significantly lower than BLEU score when special terms are translated wrong. The model first pretrained on abundant randomly sampled fake sentences and then on limited real sentences performs much better than by other training methods in translating terms correctly.

## 5. Conclusion

In this work, we have shown the effectiveness of a sequence to sequence RNN model with attention for machine translation on special fields with limited data by data augmentation. We compared the performance of training Go book sentences and Chinese cuisine menu with data augmentation to without data augmentation. The results of our experiments show that, with limited data, our data augmentation method of randomly sampling terms and connecting to make fake sentences improves the generalization performance on training set for Go book sentences, but not as well for cuisine names. It does better for sentence translation than phrase translation. We also compared our new evaluation method BLEU-POT to the commonly used BLEU score. BLEU-POT is better at measuring accuracy on translating domain-related terms than BLEU, which suits our research topic.

Future work includes research on improving Chinese word tokenization and using penalty on terms in loss function. Unlike English words, Chinese words are not separated by spaces. This presents great difficulty for translation in special area because a general Chinese word tokenization tool is not able to tokenize special terms correctly. In our work, we tokenized each character as a word for Go books translation task, and manually tokenized words for Chinese cuisine names translation task. As it is impossible to manually tokenize when doing prediction, we need better heuristics for special field tokenization.

Another thing we could further investigate in is using penalty on terms in the loss function instead of only in evaluation. We could penalize terms that are not translated correctly and back propagate the gradients on those terms. This might improve our accuracy in special field translations.

## References


[1] Chinese english parallel cuisine names. https://wenku.baidu.com/view/7cc4bc89cc22bcd126ff0c31.html. Accessed: 2018-12-17.

[2] Sensei's library. https://senseis.xmp.net/?ChineseGoTerms. Accessed: 2018-12-04.

[3] Wmt 18 news commentary. http://www.statmt.org/wmt18/translation-task.html#download. Accessed: 2018-12-17.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

[5] T. W. W.-J. Z. Kishore Papineni, Salim Roukos. Bleu: a method for automatic evaluation of machine translation. *ACL*, 2002.

[6] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *NIPS*, 2014.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin. Attention is all you need. *NIPS*, 2017.

[8] B. Zoph, D. Yuret, J. May, and K. Knight. Transfer learning for low-resource neural machine translation. *EMNLP*, 2016.