Translate English to Chinese using Seq2Seq with attention model

Zijun Wu
University of Alberta
zijun4@ualberta.ca

Ruiqin Pi
University of Alberta
ruiqin@ualberta.ca

Abstract

The exponential growth of the information recorded by different languages brings more demand for the information exchange. Thus, making machines to understand languages and translate efficiently is an important task in the neural language processing. The goal of our project would be automatic English to Chinese translation from short sequence to sequence. In the process, we will focus on an RNN based encoder-decoder translation model and compare the difference of attention and intra-attention mechanism and their performance of doing machine translation.

1 Introduction

1.1 Background

Nowadays, globalization plays a more important role around the world. The exponential growth in the information recorded by both Chinese and English brings more demand for information exchange. With deep learning development, making machines understand these two languages and do mutual translations automatically becomes an important task in the neural language processing. Some approach (Bahdanau et al., 2014a) and (Kalchbrenner et al., 2016) decide to use different kinds of attention mechanism and integrate them inside the RNN based Encoder and Decoder architecture to do the seq-to-seq translation. In contrast, (Vaswani et al., 2017) only make use of the mechanism in this task. Based on the findings above, it comes to our mind that ’attention’ plays an important role in doing seq-to-seq translation tasks. By reviewing the previous works, we found that the RNN based models with attention or intra-attention have their own good points in handling different input sequence types. It is hard to say that one of them will be obsolete completely.

1.2 Overview

This project uses RNN based encoder and decoder to do automatic English to Chinese translation in both short and long sequence-to-sequence. It compares the difference of attention (integrated with both encoder-decoder) and intra-attention (integrated with an encoder and decoder independently) mechanism and machine translation performance on our datasets.

Firstly, we need to pre-process the texts which are used for training and testing. This includes lowercasing, lemmatization, and segmentation to both of the source and target languages. Next, the following step is doing word embedding, translating words to vectors. There are a few techniques to do this, such as Word2vec or fastText. Using an encoder and decoder architecture to encode a sequence of word vectors into a fixed-length vector and the decoder part will predict the translation words. (Sutskever et al., 2014a). Furthermore, we also want to apply an attention mechanism based on the encoder-decoder model, which helps the model understand the semanteme of a sentence better. (Bahdanau et al., 2014b) The input could be a word sequence, and the output would be its corresponding translation in the proper format. For example, in English, we have ”At the start of the crisis.”, and in Chinese, the translation results would be ”随着经济危机不断加深和蔓延.” To evaluate the translation accuracy of our model, we use ”BLEU (Bilingual Evaluation Understudy)” and cosine similarity accuracy(Papineni et al., 2002) as quick and inexpensive evaluation methods for the translation in between 2 languages without the interference of humans.

2 Relative works

(Sutskever et al., 2014b) proposed a novel approach for machine translation, which implements a multilayer LSTM encoder to decoder model. Around the same time, (Cho et al., 2014) developed an RNN encoder-decoder architecture based on a novel hidden unit is simpler than the LSTM and can adaptively learn to balance between longer short-term information dependencies. The encoder RNN took a sequence of word vectors as input, and the last hidden state is output as a context feature, combined with the sequence of target word vectors fed into the decoder RNN. In their experiment, a higher BLEU score from their model indicates their approach can capture the linguistic regularities in phrase pairs and produce the well-formed target phrases better than others. However, there still exist some limitations. When training a long sequence, the context information may be lost by the nature of RNN, which would cause the performance to decrease. (Bahdanau et al., 2014a) used an attention mechanism that enables the decoder to ”look back” the hidden context from the encoder by creating context vectors that are weighted sums of the encoder hidden states. In this way, the decoder will learn to pay attention to the input sentence's most relevant word. This can lead to a big improvement in training a long sequence of words, and the overall performance will increase. However, the structure is still limited in requiring both source and target sentences. Then, (Cheng et al., 2016) proposed an intra-attention (also known as self-attention) mechanism, and its implementation is no longer limited to the encoder-decoder structure. It can be applied within the encoder itself and allow the network to learn lexical relations between the source tokens.

3 Method

3.1 Hypothesis

Referring to the statistical results in (Bahdanau et al., 2014a) and (Cheng et al., 2016), we came up with a hypothesis that the model with the help of an intra-attention module will provide better translation precision and more accurate readable contents than the model using the general attention mechanism in doing Chinese to English translation.

3.2 Dataset

There are nearly 15 million entries in The United Nations Parallel Corpus v1.0 (Ziemski et al., 2016) (English and Chinese version). So the first thing to do is to filter some entries that contain numbers and other useless punctuation except comma and period. Since the language in the dataset is quite formal, we rule that a sentence should end with a period, and any sentence without a period is also filtered. We also filter some entries in the Chinese dataset if there is any English character that appears. Short English sentences with a length less than 10 and en-zh sentences length ratio greater than 9 are also filtered. After this, there are still nearly 3 million entries remaining. Finally, we randomly pick 50k entries (a compromise for a lower running time and less memory usage) from the filtered entries to compose our dataset.

3.3 Specific procedures

Data pre-processing: We need to remove all punctuation and stop words in both English and Chinese dataset. For English, we use SpaCy (Honnibal and Montani, 2017) for removing. Moreover, for Chinese, we use the library from stopwords-zh on Github. Then we split the 50k entries with train, dev, and test dataset according to the ratio of 3:1:1. Finally, before feeding them into our model, we do the sentence segmentation into words and Word2Vec word embeddings for each sentence pair. For both English and Chinese, we use the pre-trained word embedding models from SpaCy (Honnibal and Montani, 2017). In this case, for each pair sentence of English to Chinese, we will have (n,300) and (m, 300) embedding vectors (n is the English word tokens embedded by Spacy, m is the embedded Chinese word tokens.). We will feed pairs of word embedding sequence as input to our models.

Algorithm and Experiment: We build two models in general. The first one is the RNN encoder-decoder model with an attention module applied to decoder (Bahdanau et al., 2014a). The second one is the RNN encoder-decoder model with an intra-attention module applied in both encoder and decoder (Cheng et al., 2016).

The training procedure of the first model takes each pair of embedding vectors and feeds the source vectors one by one into the encoder part. Next, the final hidden vector in the encoder is used as the decoder part's initial hidden state. The encoder output and the attention weight calculated by the attention module will both be regarded as decoder input. In the first iteration, the decoder part takes a start token as input, and in each time step, the output vector of the decoder and the attention weights are considered as inputs in the next time step. The output vectors are compared with the target vectors one by one through Cosine Embedding loss. Finally, once the decoder predicts the entire sentence, it back-propagates the loss and goes for its next training pairs.

The training procedure of the model with intra-attention is quite different. Since the intra-attention is used independently in both the encoder and decoder, the first intra-attention module takes the embedding vectors as inputs. The encoder will calculate the hidden vectors based on both intra-attention weights and embedding vectors. The second intra-attention module will process the hidden vectors from the encoder first and use the second intra-attention weights as the initial hidden state of the RNN decoder. Moreover, We also modify the intra-attention module to read a token one by one, not the entire sentence as a whole, in order to meet the same training setting.

In this case, the validity is ensured since the input and output of the two models are in the same shape; thus, they can be evaluated in the same way. The first statistical testing result is based on the average cosine similarity between the predicted sequence and the target sequence. The second one is based on the BLEU score, which needs us to translate the predicted vectors into a sequence of words. For each predicted vector, we find the closest word in the embedding space using cosine similarity. Moreover, to ensure reproducibility, the code of our algorithm and a user guide to experiment also be explained in very detail on GitHub.

4 Experiment and result

For the experiment settings, both models are compared under the same dataset, and the training losses are recorded. The training parameters are controlled to make sure that both models are trained under the same circumstance. The evaluation metrics are ”BLEU” and Cosine similarity score, and the predicted sequences and the target sequences will be compared by calculating these 2 scores. If we notify evident higher scores for the intra-attention model in two experiment metrics, we can say that our hypothesis is correct.

By referring to the figure 1. In the current experiment stage, we recorded the training loss for different models. It is important to emphasize that all losses start around 1, and then go down (*graph 1 does not contain the initial loss in this stage, the first loss value is recorded at iterate 1000.*) Here,

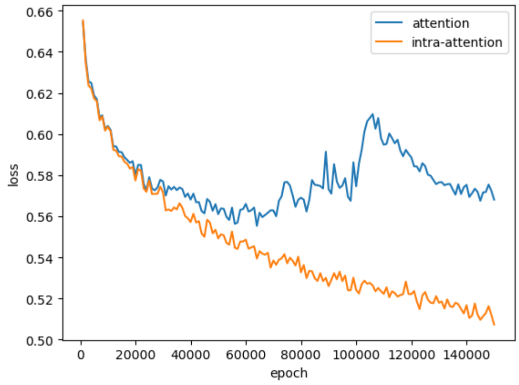

Figure 1: The training loss for 3 different models. Blue: RNN with attention module; Green: RNN with intra-attention module; Yellow: RNN with intra-attention module and Convolutional layers

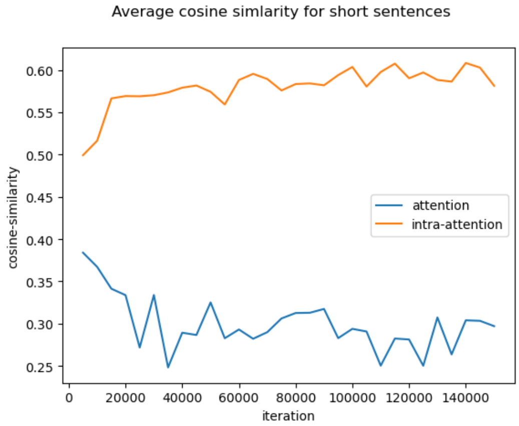

Figure 2: The average cosine similarity for the embedded vectors of 30 short sentence. Yellow line is for the model with intra-attention module, blue line is for the model with general attention mechanism

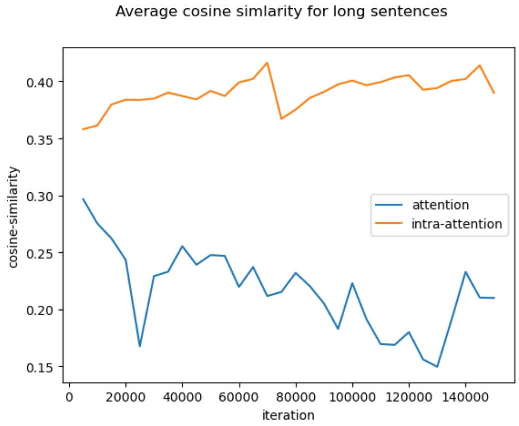

Figure 3: The average cosine similarity for the embedded vectors of 30 long sentence. Yellow line is for the model with intra-attention module, blue line is for the model with general attention mechanism

we trained 3 models in 150000 iterations with 30k random choose paired embedding vectors. The RNN model with the attention module started to converge very well but encountered a very serious over-fitting and oscillating in around 0.56. Instead, we could observe a perfect convergence on our modified RNN with the intra-attention module throughout the whole time. Even at very late iterations, it still has the potential to converge better. In this case, the loss statistics indeed provided a trend that the intra-attention module could provide lower error when doing the translation.

BLEU	Attention	Intra-attention
short-sentence	14.78	16.8
long-sentence	3.41	8.96

Table 1: Comparison between seq2seq RNN model with Attention and Intra-attention

Referring to the table 1 and the figures 2, 3, we noticed that the average BLEU score and cosine similarity score are evidently higher amongst short sentences (with around 10 words) than in the long sentences (greater than 15 words). Moreover, for both short and long sentences, the intra-attention module model is more robust to overcome the over-fitting problems and could provide a higher average BLEU score and cosine similarity score than the model with the attention mechanism.

5 Discussion

5.1 Compare and contrast

(Bahdanau et al., 2014a) propose the translation model with an attention mechanism. They show that their BLEU scores are very stable as there is no degradation on the sentences with less than 60 words in the WMT14 English-French dataset. When using the same model and performing on our dataset, the degradation becomes quite evident. The shorter sentences with about 10 average words have an average similarity around 0.3-0.4 and an average BLEU score at 22. For the longer sentences with about more than 15 words, the average similarity is around 0.2-0.3, and the average BLEU score at 12.5. The reason that results in the difference between our findings and the results previously reported could be the less training data: We pick 50k pairs of the sentences, much less than the 384M training pairs used in the paper mentioned above. Then it would be more common for our model to encounter the unseen words or sentence structures, which will cause a reduction in the model performance.

In the paper(Cheng et al., 2016) propose the model with the intra-attention module. They did not examine their model's performance in seq to seq translation. Nevertheless, when we apply the intra-attention in the seq to seq model, it could obtain a higher similarity score and BLEU score than the model with an attention mechanism. Since we train both models on the same dataset with the same iterations, it can be said that the intra-attention module improves the seq-to-seq translation accuracy.

5.2 Error Analysis

One reason that causes a very evident error between our model translated results and the target results could be the too small embedded vector size. Each word is encoded into a 300 float vector. It is easy to find another similar vector, but with a complete unsimilar meaning by calculating the cosine similarity between them. This is the main reason why these two models can produce better cosine similarity scores than BLEU scores. For example, the word token ”今天(today)” and the token ”这两天(these two days)” have a cosine similarity of 0.92 but with completely different meanings. Thus, the embedding size 300 may not be robust enough to handle the representation of a huge amount of words in Chinese. If we want more accurate predictions, the dimension of word embeddings have to be large enough to avoid mismatching problems. Moreover, the artificial error of encoding similar Chinese words into very non-similar vectors would also be a reason for the errors. For example, the word embeddings for ”军队(army)” and ”部队(military)” has the cosine similarity 0.67, but they are the synonym. One example of failed translation is, ”I give the floor to Mr. Prins” should be translated to ”我请詹姆斯先生发言” but our model predicts ”他当时建议” which means ”He suggests that time.” Therefore, if the model cannot predict the exact same vectors as the target word vectors, the errors made by the models can be amplified and reflect on the translation results.

5.3 Limitation

Although our model is designed to translate between English to Chinese, it is trained by feeding English sentences and outputting Chinese sentences: a single direction training over two languages. So it is impossible to translate Chinese into English without re-training the whole model by reverting the source and the target languages. Because of the large data set and the non-parallel RNN, a re-training is quite time-consuming and inefficient.

6 Conclusion

In conclusion, by re-implementing the RNN based encoder and decoder architecture with the attention and the intra-attention mechanisms, we can make the machine learn translation between English and Chinese. For most short sentences, both BLEU score and Cosine Similarity accuracy have shown that we could translate them in relatively higher precision than in long sentences. Moreover, in our contrast experiments, the evaluation metrics also show the model with the help of intra-attention outperforms the model with the attention mechanism in the translation task. However, drawbacks still exist in our approach. Future improvements in data pre-processing and model structure are needed to demonstrate a more robust translation result.

7 Responsibilities

Zijun is in charge of processing the data, and Ruiqin is in charge of building the translation models. Both are work together for training and tuning the models and summarizing the statistical results.

8 Acknowledgments

We want to thank Denilson Barbosa, Carrie Demmans Epp, Greg Kondrak, and Peiran Yao for helpful discussions and feedback.

9 Github

https://github.com/khalilbalaree/machine-translation

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014a. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014b. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014a. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014b. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534.