Neural Machine Translation Using Recurrent Neural Network

Jian-Syuan Wong

October 2017

1 Definition

1.1 Project Overview

Because of the advancement of internet technology, the interactions and communications of people across the world have been greatly enhanced. In addition, the users are enable to easily perceive information from multiple information sources all over the world. Therefore, the needs of language translation have been significantly increased as well. Various online services has been introduced to help users understand text in different languages such as Google Translate which translates more than 100 billion words a day[1]. The recent approaches leverage deep learning and representation learning to train and build a sequence-to-sequence (seq2seq) neural machine translation model [2, 3].

1.2 Problem Statement

Due to the increasing needs of language translation services, the objective of this project is to build and train an end-to-end neural machine translation model

[1] https://www.blog.google/products/translate/ten-years-of-google-translate/

that can be applied to translate language in English to French. Unlike the traditional approaches, the knowledge in certain languages such as grammars and the relevant linguistic information won't be required. This model should be able to read the input data (text in English) and generate the desired output (text in French).

The seq2seq model [11] was built with Tensorflow. it involves two major components Encoder and Decoder (see Figure1 for illustration - *from Deep Learning for Chatbots : Part 1*[2]). The encoder is a recurrent neural network (RNN) used to learn a fixed dimensional vector representation from the input data (e.g., sentences in English). The decoder is another RNN that can be applied to take the vector representation (the last hidden state) from the encoder as input and decode the information into the text in target language.

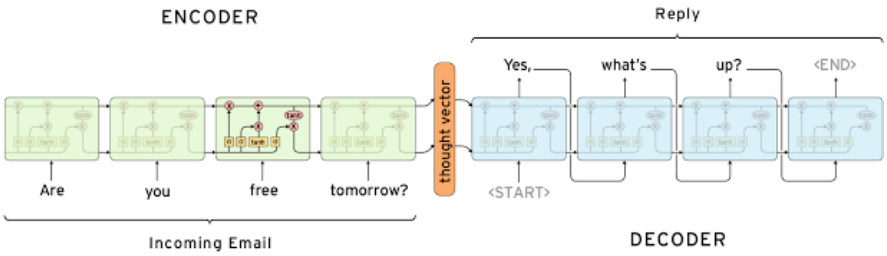

Figure 1: The sequence-to-sequence model

1.3 Metrics

First, the changes of losses for both training data and validation data will be stored and displayed during the training process. From the losses in different iterations, we are able to understand whether the model was trained properly. To evaluate the effectiveness of the neural network model for machine translation tasks, BLEU (bilingual evaluation understudy) score will be calculated using

[2]http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/

NLTK[3]. BLEU [9] is introduced by Papineni et al. in 2002 to assess the quality of language translation since "*Human evaluations of machine translation are extensive but expensive*". The BLEU score is ranged from 0 to 1, and the larger score indicates a better translation quality. Natural Language Toolkit (nltk.translate.bleu_score.sentence_bleu) was leveraged to calculate the sentence-level BLEU score.

2 Analysis

2.1 Data Exploration and Exploratory Visualization

The data[4] used for this project is provided by Udacity for the capstone project (NLP concentration) of Artificial Intelligence Nanodegree. There are 137,860 pairs of sentences are in both English and French. The sentences in English will be used as the input data and the sentences in French will be utilized as the output labels. In this project, The original dataset is split into training and test data - 1 percent of the dataset (1379 sentences) is used for model evaluation.

An example of the dataset is listed below:

- **English:** new jersey is sometimes quiet during autumn , and it is snowy in april.
- **French:** new jersey est parfois calme pendant l' automne , et il est neigeux en avril.

Additionally, the length of sentences in both English (input) and French (target) are also investigated. As shown in Figure 2, the maximum length of input sentences is less than 20 words, and most of the input sentences have the length around 15 words. Moreover, the maximum length of target sentences is

[3]http://www.nltk.org/
[4]https://github.com/udacity/aind2-nlp-capstone/tree/master/data

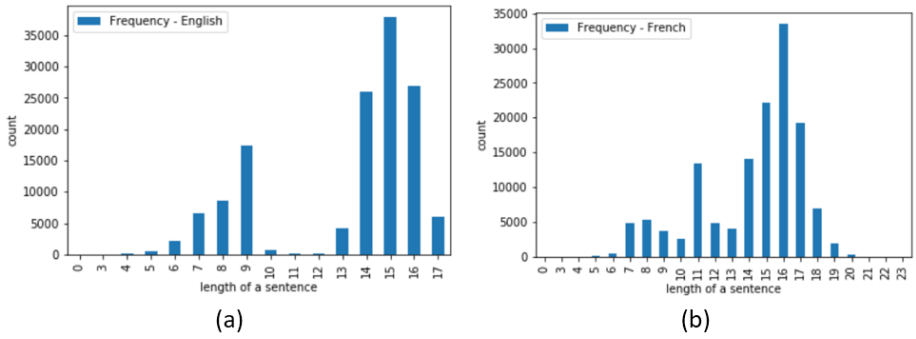

Figure 2: The length distribution of (a) the input data and (b) the target data.

less than 25 words, and most of the input sentences have the length around 16 words.

2.2 Algorithms and Techniques

2.2.1 Recurrent Neural Network

Recurrent Neural Network (RNN) [7] is one of the popular artificial neural network. Because of the internal memory mechanism, RNN is feasible to process sequences of inputs. Various natural language processing(NLP) tasks such as sentiment analysis and text summarization can be accomplished with RNN. Therefore, RNN is utilized in this project to build the seq2seq model.

2.2.2 Long short-term memory

The simple recurrent network often encounters the issue of vanishing and exploding gradients when performs backpropagation for model optimization [5]. To resolve the issue of vanishing gradients, long short-term memory, a RNN architecture, is commonly utilized. The architecture of LSTM is shown in Figure 3 (from *Understanding LSTM Networks*[5]). Different gate units in LSTM are

[5]http://colah.github.io/posts/2015-08-Understanding-LSTMs/

used to determine the information to be added, retained or forgot.

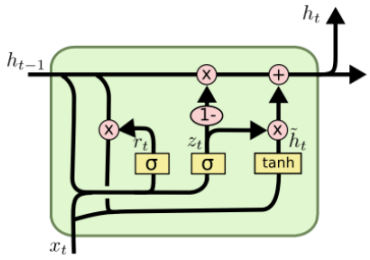

Figure 3: The Long Short Term Memory architecture

2.2.3 Sequence-to-Sequence model

A Sequence-to-Sequence model is often leveraged to build a model for language translation and chat bot development. The model comprise two major components, the encoder and the decoders. Through the encoder-decoder mechanism, the model is able to learn the representation of input sequences with the encoder and generate the desired output sequences with the decoder. In contrast to the traditional statistical machine translation models which require the domain knowledge in the involved languages, seq2seq model is relatively intuitive to ed built and trained

2.3 Benchmark

A vanilla seq2seq model with will be implemented as a baseline model. The final model would adopt additional techniques to improve the performance. The improvements include exploiting a bidirectional RNN for the encoder, and leveraging the attention mechanism to facilitate the decoding process.

3 Methodology

3.1 Data Preprocessing

1. The first step is to ensure all of the words is in lower case since a word in either upper case or lower case would be considered as the same word.

2. In this dataset, the input sentences and target sentences have different length. To perform mini-batch training we need to pad ("⟨PAD⟩") both the input sentences and target sentences to ensure the input sentences have the same length in a batch as the same for target sentences.

3. "⟨GO⟩" and "⟨EOS⟩" are prepended and appended to each target sentence, respectively. Those tags help the model to understand the start and the end of a target sentence. Such tags are important for both training and inference purpose.

4. The text is then converted into the corresponding ids (one-hot-encoding) since most of the machine learning libraries cannot take text type of data as the input.

5. To obtain the better representation of a word, word embedding [8] is applied to transform a word into a vector. Word embedding has the capability to capture the syntactic and the semantic meaning of a word. A trainable matrix (with the shape [vocabulary_size, embedding_size]) was created to convert a word into the vector representation. *vocabulary_size* indicates the number of unique terms in a corpus and *embedding_size* implies the size of a word vector. Since the dataset contains sentences in two different languages (corpora), two embedding matrices are used to convert English words and French words into the corresponding word vectors.

3.2 Implementation and Refinement

The project is implemented based on the framework offered for the language-translation project[6]. For the project implementation, I also learned a lot from the neural machine translation (seq2seq) tutorial[7].

A seq2seq model typically comprised two main components, a RNN as a encoder and and another RNN as a decoder. To build a RNN, Long short-term memory (LSTM) and Gated recurrent unit (GRU) are commonly used. Both architecture can effectively mitigate the issue of vanishing gradients, therefore, a RNN model built with LSTM cell and GRU cell can still have a decent performance with a longer sequence data. Additionally, dropout [10] is utilized to prevent the network from overfitting. According to [1], LSTM outperforms GRU for the machine translation model. Therefore, I choose LSTM for the RNN implementation. In this project, TensorFlow[8], an open-source software library for machine learning , is applied to build and train the neural machine translation model.

Encoder

I employed a multilayer bidirectional RNN for the implementation of the encoder since the bidirectionality on the encoder often produces the better performance. The main difference between a unidirectional RNN and a bidirectional RNN is that a bidirectional RNN maintains two hidden state, one for left-to-right propagation and the other for right-to-left propagation (see Figure 4 - from *A trip down long short-term memory lane*[9]). The bidirectionality offers the capability for predicting the next word through summarizing the past and future information.

[6] https://github.com/udacity/deep-learning/tree/master/language-translation
[7] https://github.com/tensorflow/nmt
[8] https://www.tensorflow.org/
[9] http://www.cl.cam.ac.uk/ pv273/slides/LSTMslides.pdf

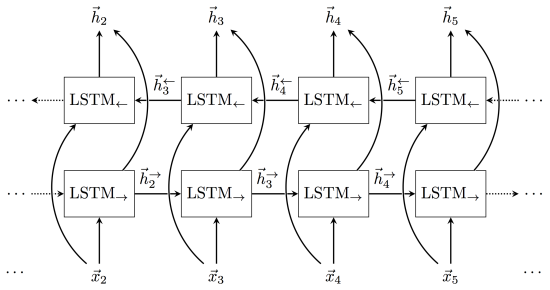

Figure 4: The sequence-to-sequence model

Decoder

Two types of decoders were built for different purpose, one for training and the other for inference.

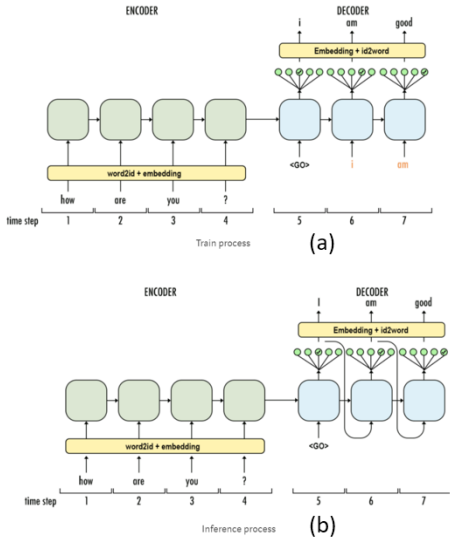

Figure 5: Two types of decoders (a) decoder for training and (b) decoder for inference

The training decoder is used for training purpose, and it takes the target sequence as the input to facilitate the training process (see Figure 5(a) - from

Seq2Seq intro[10]). The inference decoder is exploited to generate the translated sentence, and the output of previous timestep is utilized as the input of current timestep Figure 5(b). A densely connected layer is added on the top of the decoders to generate the predicted word in each timestep. To improve the performance, the attention mechanism [6] is utilized that allows the decoders to obtain the most relevant memories from the encoder to enhance the decoding process.

The parameters for building and training the model is shown in Figure 6.

```
# Number of Epochs
epochs = 5
# Batch Size
batch_size = 128
# RNN Size
rnn_size = 64
# Number of Layers
num_layers = 2
# Embedding Size
encoding_embedding_size = 64
decoding_embedding_size = 64
# Learning Rate
learning_rate = 0.001
# Dropout Keep Probability
keep_probability = 0.8
```

Figure 6: Parameters used for the project implementation.

Loss and Optimization

tf.contrib.seq2seq.sequence_loss function is applied to calculate the weighted cross-entropy loss for a sequence of logits. While calculating the the loss, a mask is used to exclude the padding part. The padding was only added to ensure the sequences in a batch have the same length, thus, the padding part should be excluded from the loss calculation. To optimize the model, the Adam optimizer [4] is utilized. Moreover, the gradient clipping method is used to diminish the problem of exploding gradients.

[10]https://medium.com/@Aj.Cheng/seq2seq-18a0730d1d77

4 Results

4.1 Model Evaluation and Validation

To ensure the model was built and train correctly, I plot the training loss to evaluate whether the model works properly. As can be seen in Figure 7, the training loss of both the baseline model and the final model decreased constantly, implying the models were built and trained the properly. Additionally, the final model (Figure 7(a)) was converged earlier than the baseline model (Figure 7(b)).

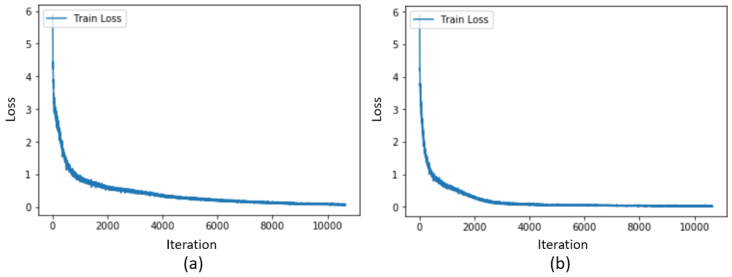

Figure 7: Training loss of (a) baseline model and (b) final model.

To evaluate both models, they were applied to translate the testing data. The BLEU scores were calculated based on the translated result. The BLEU score for the baseline model and the final model are **0.798** and **0.724**, respectively.

The results indicate the baseline model has the better performance than the final model. The possible reason leads to the result could be that the length of the sentences in the dataset is relatively short (most of the input and target sentences have the length around 16 words), therefore, the bidirectional RNN and attention mechanism might not be helpful.

4.2 Justification

The objective of this project is to build a machine translation model that is able to translate English sentences into. Additionally, the background knowledge and expertise in both languages should not be required when build and train the model. To achieve the objective, a seq2seq model was built and trained. Based on the result of model evaluation using test data (the BLEU score is higher than 0.7), I would consider this model should be able to perform the translation task appropriately. As the translated sentences listed in the *Free-Form Visualization* subsection, most of the translated sentences matched the target sentences well.

5 Conclusion

5.1 Free-Form Visualization

To demonstrate the trained neural machine translation model can be utilized to translate sentence English into French. I randomly printed out the results when evaluating the model using the test dataset.

The results are listed below:

- English Words: your least liked fruit is the pear , but their least liked is the banana .
- French Words (translated): votre moins aimé fruit est la poire , mais leur moins aimé est la mangue .
- French Words (target): votre moins aimé fruit est la poire , mais leur moins aimé est la banane .

- English Words: new jersey is quiet during january , but it is sometimes rainy in fall .

- French Words (translated): new jersey est calme en janvier , mais il est parfois pluvieux en février .
- French Words (target): new jersey est calme en janvier , mais il est parfois pluvieux à l' automne .

- English Words: california is usually busy during spring , but it is never rainy in january .
- French Words (translated): californie est généralement occupé au printemps , mais jamais des pluies en janvier .
- French Words (target): californie est généralement occupé au printemps , mais jamais des pluies en janvier .

- English Words: the apple is my most loved fruit , but the grape is his most loved .
- French Words(translated): la pomme est le fruit le plus mon cher , mais le raisin est le plus aimé
- French Words (target): la pomme est le fruit le plus mon cher , mais le raisin est le plus aimé .

5.2 Reflection

In this project, I applied TensorFlow to build and train a seq2seq neural machine translation model. Unlike, traditional statistical machine translation approach that relies on the aids of language experts to tweak different components of a model (e.x., how to correctly align the input language and translated language), seq2seq is relatively simple. This project took me a significant amount of time

to understand the details of how seq2seq works (e.g., LSTM, RNN, encoder, decoder, etc) and the techniques for improvement such as bidirectional RNN and attention mechanism. I gained a lot of related knowledge of RNN and hand-on experiences on TensorFlow.

5.3 Improvement

There are some potential directions for future improvement. First, a larger training data can be utilized. In this project, I only used around 130,000 sentences to train this model. The state of art machine translation model often used millions of sentence for training. However, it requires a significant amount of time to train a model with a large dataset. Additionally, the beam search technique can be utilized to select the better candidate. For the decoding process in this project, a greedy approach is applied to select the best output at each time step. Nonetheless, if a mistake made in early step, it may cause a significant impact for the entire decoding process. Beam search technique often yields the better performance by mitigating such concern by retaining multiple candidates at each time step.

References

[1] Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*, 2017.

[2] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[3] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *EMNLP*, volume 3, page 413, 2013.

[4] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[6] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[7] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

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

[10] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

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