Project: Contrastive learning for machine translation

Zilong Wu, Jianing Chen and Xuchen Hu (Group #4)
Electrical and Computer Engineering Department
New York University
Brooklyn, NY 11201
{zw2599, jc10034, xh2090}@nyu.edu

Abstract

This research aims to propose a new architecture that improves the performance for distant language translation. We design a contrastive learning based model for Japanese to English translation. The principle idea behind the model is to make the embedding of Japanese sentences similar to the relevant English sentences thus the backbone model would treat the Japanese embedding same as English. We add a multilayer perceptron on the Japanese embedding to mimic the format of English embedding and learn the rule of English through the contrastive learning. The new designed model shows good performance in Japanese to English translation.

1 Introduction

Translating distant languages is a tough problem for machine translation. Distant language means that the structure of two different languages have huge divergences and the representations of embedding could differ greatly. For example in Japanese to English translation "りんごを食べたい—I want to eat an apple". The sequence of those two sentences is totally different, the "apple" which should be translated in the front of the sentence dropped at the end. Instead, the subject 'I' which is omitted in the Japanese appears at the very start position in the English translation. These kinds of gaps between two languages are obstacles for neural machine translation. How to overcome this problem is researched in this project.

Neural network models have made great achievements in natural language processing, transformer model [1] which was proposed in 2017, utilizes vanilla attention mechanisms in neural networks to fetch features. With the introduction of transformer, more and more powerful language models have been proposed in natural language processing. Shortly after the invention of transformers, Bert [2] moves one step forward. It incorporates the encoder of the transformer to predict the masked vocabulary in the input sentences. Bert achieves shocking performance by leveraging a masked language model over transformer model and advances state of the art performance over eleven natural language processing tasks. However, with those powerful language models, the BLEU score [3] with distant languages is not as high as similar languages such as France and English. In this project, we propose a contrastive learning based model to improve the performance of distant language translation.

Contrastive learning is a popular form of self-supervised learning that encourages augmentations of the same input to have more similar representations compared to augmentations of different inputs. Contrastive learning aims to learn effective representations by pulling semantically close neighbors together and pushing the non-related neighbors away from each other. This architecture was first used in computer vision for visual representation, encouraged by the success of contrastive learning in computer vision, researchers in natural language processing started to apply this structure in text classification and other tasks in natural language processing. SimCSE [4], a simple contrastive learning of semantic embedding which predicts input sentences itself with dropout used as noise. By inputting the sentences into dropout pretrained encoder twice, SimCSE acquires two different embeddings as positive pairs. Then other sentences from the same mini-batch are treated as negatives

and the language predicts the positive pairs among the negative sentences. This simple architecture outperforms the next sentence. Inspired by SimCSE, this project constructs a contrastive architecture for machine translation.
In this project, we make two main contributions to distant language translation.
(1) We incorporate the contrastive learning in machine translation. The source language sentence and the corresponding target language sentence are treated as augmentations of same input. The objective of contrastive learning is to make the embedding of the input pair have similar representations compared to different source sentences.
(2) We utilize a Multilayer perceptron (MLP) to mimic the source language sentences to the target language sentences. In this case, we only need to train paramters in MLP.

2 Related work

In this section, we would generally review some approaches related to machine translation and contrastive learning.

2.1 Tokenization

Tokenization is an indispensable step in natural language processing (NLP) tasks. Byte-Pair-Encoding (BPE) [5] is a popular method to split words into pieces based on how frequently they appear. The main idea of BPE is to compress the words into smaller subwords and decompose rare words into commonly appear subwords. BPE is an efficient strategy to deal with rare words and open-vocabulary.

2.2 Sequence to Sequence learning

In machine translation both the input and output are a variable-length sequence. To address this type of problem, Sequence to Sequence learning (Seq2Seq) [6] is proposed to convert sequence from one domain to another. The encoder-decoder architecture allows the input and output have different length sequence.

2.3 Transformer model

Transformer [1] is a popular language model for natural language processing tasks. The main idea of Transformer model is to eliminate the long term dependency in previous neural networks such as long short term memory (LSTM) [7]. Transformer model utilizes only attention mechanism to build the neural networks and allows the model to learn all knowledge from all positions of sentences. In machine translation, transformer incorporates the Seq2Seq learning and build the encoder-decoder structure. The difference between the transformer mode and the previous Seq2Seq model is that transformer uses the attention mechanism while the former model uses recurrent neural networks.

2.4 Contrastive learning

Contrastive learning is a form of self-supervised learning that encourages augmentations of the same input to have more similar representations compared to augmentations of different inputs. The purpose of contrastive learning is to distinguish the features of different inputs, that the original input and augmented input would learn to have higher score than different input with the augmented input. In this project, the original input and the augmentation becomes the source language and the target language. The the pair of source sentence and target sentence would have higher similarity while the different source sentences with the given target sentence would have lower similarities.

2.5 dataset

In this project we use a Business Scene Dialogue (BSD) [8] dataset for translation, a Japanese-English parallel corpus containing written conversations in various business scenarios. The dataset was constructed in 3 steps: 1) selecting business scenes, 2) writing monolingual conversation scenarios according to the selected scenes, and 3) translating the scenarios into the other language. Half of the monolingual scenarios were written in Japanese and the other half were written in English.

3 Model architecture

In this section, we would explain the whole architecture of our model and the main idea behind this model.
The neural machine translation process could be viewed as a simple function $y = F(x)$, where y is the output and x is the input word embedding from a different language. F represents the language model which catches semantic information from source language and translates it to the results.Suppose the input language is the same as the output language, x is y' s embedding vector and the output of the language model should perfectly match the target sentence. Thus if there is a function g that makes $g(x)$ approximately equal to y' s embedding, the performance of the language model should be improved. This proposal builds a contrastive learning architecture to fulfill the approximation concept.

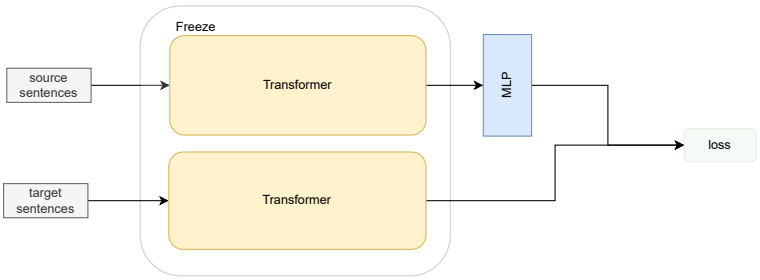

Figure 1: Architecture of contrastive learning for contrastive learning

Figure 1 shows the architecture of contrastive learning, source language and target language are inputted into the same freezed transformer encoder. The difference between them is that the source sentences are modified by a multilayer perceptron (MLP) layer which reshape the source embeddings to 'target embeddings'. The reason why we add a MLP is that we want to mimic the vector of target sentences based on the source sentences. By training the parameters within MLP layers, the MLP would learn how to transform the source vector to target vector. The position of MLP layer could be placed at the output of target encoder, however, since we want to transform the source embedding to target, the MLP layer would not be placed on target side.
In training process, the input source sentence and the related target sentence are positive pairs while other sentences in the same mini-batch are negatives. To distinguish the positive pair and negative pairs, this project uses vector similarity to represent loss function. Positive pairs would have higher similarity scores and the other pairs would be trained to have lower scores. Thus the MLP would learn better to mimic the target language embedding.

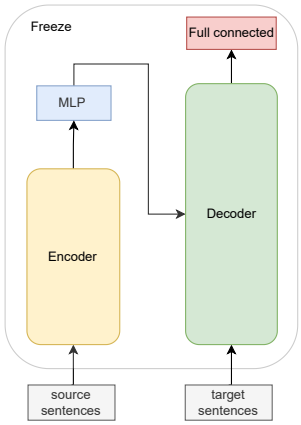

Figure 2: Architecture of transformer based translation model

Figure 2 shows the entire translation process. In contrastive learning, we only train the parameters in MLP layers, the other parameters in transformer model do not change. In the translation process, the MLP would keep the parameters learned from contrastive learning and transform the original encoder embedding to 'target' encoder embedding. During the whole translation process, no paramters would be updated since we only mimic the source language to the target language.

4 Initial attempts

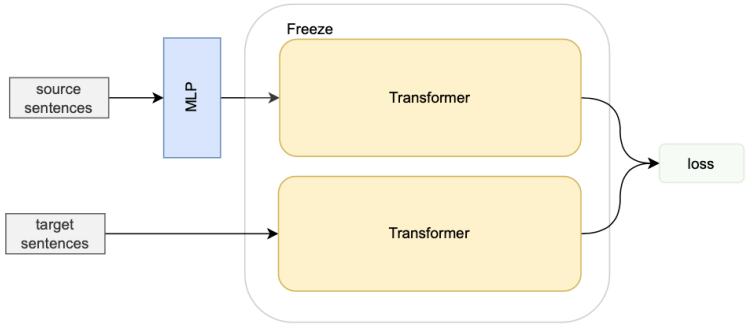

Figure 3: Architecture of initial attempt

At the very beginning of the project, we decide to put the multilayer perceptron (MLP) layer in front of the transformer model. Figure 3 shows the initial attempt, the MLP layer would transform the word embedding of source sentences to the 'target embedding'. However, in the later learning of machine translation and transformers, the initial embedding commonly do not use word2vec [9] or Glove [10]. The word vectors generated by those embedding methods are hard to use in machine translation. The reason of this is that machine translation needs to translate one language to another, the complicated word embedding would be difficult for language model to translate the sequence back to individual words. Thus the initial embedding should not be that complicate. Under this circumstances, the MLP layers would not be neccessary.

Besides the structure modifications, we change the dataset of this project. We tend to use the OPUS open corpus dataset at first. Unfortunately, the OPUS only has English to Japanese dataset. We change the dataset to Business Scene Dialogue which translate from Japanese to English. The direction of source language and target language is important, since we need to compare the ability of translation from the native language to another language.

5 Following questions

There are still multiple problems need to solve. (1) The Transformer based encoders have multiple encoder blocks, which means we may need multiple MLP layers to transform the source language to the target language. Then how to calculate the similarity of target sentence embedding and the source language embedding becomes a problem for us. (2) Since the pretrained models commonly combined the encoder and decoder together, how to insert the MLP layers into the composed language model is another question we need to solve.

6 Contemporary results

6.1 Clean dataset

We extract the Japanese sentences and correlated English sentences out of the raw datasets and tokenize them through different tokenization method. For Japanese vocabulary, we use the fugashi [11] to process the Japanese sentences. For English sentences, we just lower the case of words and split them with space. The over all tokens lengths are shown in figure 4. Since most of sentences compose less than 20 words, in the experiment procedure we would chose 20 as number of steps.

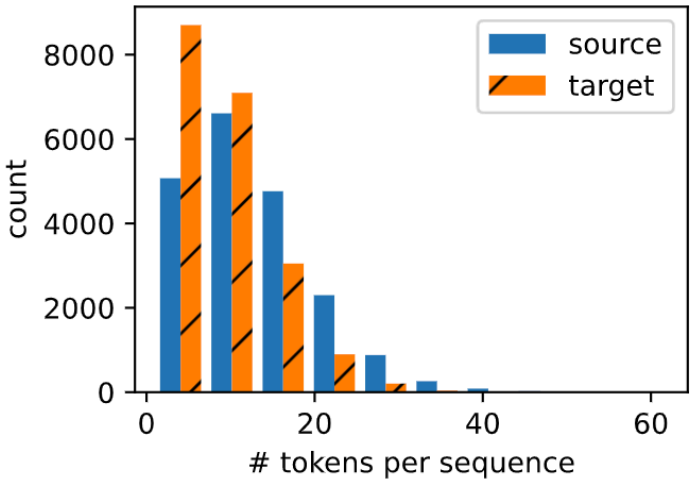

Figure 4: Length for Japanese sentences and English sentences

6.2 Initial experiment

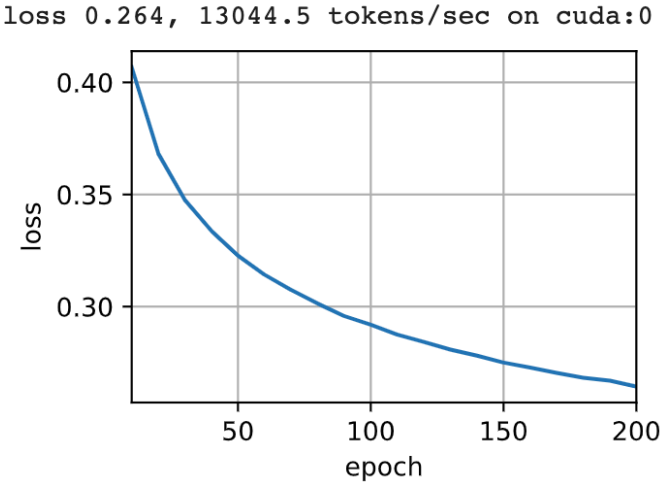

Figure 5: Loss of the simple transformer model

```
ウェイン、調子はどうです？ => so what is it?, How is it going, Wayne? bleu 0.000
まあまあです。 => i'm not too <unk>, I'm not too bad. bleu 0.537
今日はご足労ありがとう。 => thank you for coming today., Thank you very much for coming out today. bleu 0.347
景気はどうです？ => so what kind of <unk>, How's business lately? bleu 0.000
おかげさまで、順調です。 => wow, that's extremely difficult, and really really really really really, It's been good. bleu 0.000
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().
```

Figure 6: Translation effect

At the beginning of the experiment, we manually construct a simple transformer model for translation. Since we need to clarify the effect of transformer model, we build a two layer encoder-decoder transformer to do the experiment. The performance of the simple model could be viewed from figure 5 and figure 6, the model could generate simple translations. The bleu scores of the translation are not good enough. More structure refinements need to be made in the following work.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] 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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[4] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[5] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] Matīss Rikters, Ryokan Ri, Tong Li, and Toshiaki Nakazawa. Designing the business conversation corpus. In *Proceedings of the 6th Workshop on Asian Translation*, pages 54–61, Hong Kong, China, November 2019. Association for Computational Linguistics.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[10] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[11] Paul McCann. fugashi, a tool for tokenizing japanese in python. *arXiv preprint arXiv:2010.06858*, 2020.