Chinese-to-English machine translation

Cynthia Hao
Department of Computer Science
Stanford University
chao16@stanford.edu

Zhuoer Gu
Department of Computer Science
Stanford University
guzhuoer@stanford.edu

Abstract

The ability to automatically translate different languages from images or handwritten text has many applications in medicine, travel, education, international commerce, text digitization, and many other areas. However, the different grammar and lack of clear word boundaries in Chinese presents challenges when translating to word-based languages such as English. In this work, we have implemented a deep learning machine translation system to tackle these challenges. The deep learning algorithm takes in Chinese text as input and uses a sequence-to-sequence (seq2seq) encoder-decoder model with an attention mechanism based on Google's Neural Machine Translation (NMT) model to translate the text to English output [8]. The model was trained using sparse categorical cross entropy loss and an Adam optimizer on paired Chinese and English text sentences from the 2019 Conference on Machine Translation, with 227,177 training pairs and 2,002 validation pairs [1]. In addition to tracking loss over training epochs, we measured the quality of our model's translations using the BLEU score for machine translation. We compared the model's performance to a smaller baseline model with no pre-trained embeddings, as well as several less complex models with different learning rates. Our final model achieved a maximum BLEU score of 0.247. We can further improve this score by tuning other hyperparameters and increasing the complexity of our model, as well as by training on a larger subset of the data to avoid biased results.

1 Introduction

Machine translation systems have applications in fields ranging from travel to the study of ancient literature. Translation applications are so important that Google, WeChat, Microsoft, and many other companies have all implemented their own machine translation systems using various deep learning methods. Mandarin Chinese is currently one of the most widely-used languages in the world, and the ability to understand Chinese text can lead to cultural and commercial benefits for non-Chinese speakers. The goal of our proposed system is to translate Chinese text into English text, one sentence at a time, to make these benefits more accessible to English speakers. In our project, we intend to build a smaller translation model based on models built by Google Translate.

2 Related work

The state of the art approach for machine translation is the system used by Google Translate. Google previously used the Google neural machine translation (GNMT) system with an 8-layer LSTM encoder and RNN decoder architecture with attention, which achieved very good performance [8]. However, it would be computationally expensive and slow to train, especially for a team without Google's resources. Google later changed their machine translation system to a transformer model

CS230: Deep Learning, Winter 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

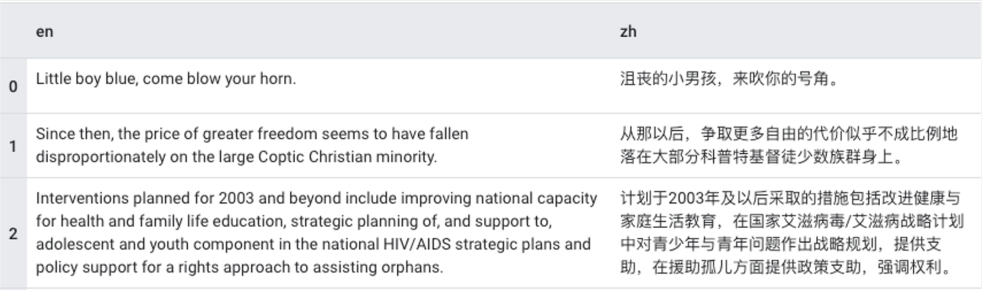
with self-attention using the Tensor2Tensor library, which is faster to train and performs better, but is a more complex model that is slightly harder to understand conceptually [5]. A team from Microsoft has also applied dual learning models to the machine translation task with success [7]. There are many existing datasets and tutorials utilizing Google's original NMT architecture because it is so well-known. We will be adapting one of these existing models for our purposes [6].

3 Dataset and Features

For our machine translation task, we used subsets of the Chinese to English news text dataset from the 2019 Conference on Machine Translation (WMT), which is freely available on TensorFlow as part of their larger wmt19_translate dataset [2]. This sentence-level dataset contains news in both English and Chinese languages in TensorFlow text object format. The full dataset contains 25,986,436 training and 3,981 validation examples [2]. Because of limitations on computational resources and to speed up training, we used subsets of the data. We further split the data into a training set of 227,177 sentence pairs and a development set of 2,002 pairs. An example of the data is shown in Figure 1.

To preprocess our data, we generated 2 separate files containing only the English sentences and only the Chinese sentences in the same order. We were able to generate this vocabulary by tokenizing each sentence in the training dataset and keeping only unique word or character tokens, and then mapping these tokens to an index in the vocabulary. We tokenized the English sentences on a word level. For our baseline model, we split the Chinese sentences by character, which may have resulted in loss of some of the semantic meaning of the Chinese words, compared to splitting into Chinese words or phrases. In our final model, we instead used the Stanford Chinese Word Segmenter, which splits Chinese text into semantic groups rather than just by character [4]. After we generated these tokens, we wrote them into a separate vocabulary file with one token per line.

In our baseline model, we chose not to use an existing embedding technique such as word2vec. The embeddings were passed directly into the rest of our model, as described in Methods. For our final model, we used the FastText word vectors for 157 languages as embeddings for both the English and the Chinese sentences [3]. Although the default embedding size is 300 units long, we reduced the size of the embedding to 64 units per vector to conserve system memory. After we generated one vector embedding for each token in our vocabulary files, we created embedding files with one token and its corresponding vector embedding on each line that were fed into the rest of our model. More details about the baseline model will be provided in the following Methods section.



	en	zh
0	Little boy blue, come blow your horn.	沮丧的小男孩，来吹你的号角。
1	Since then, the price of greater freedom seems to have fallen disproportionately on the large Coptic Christian minority.	从那以后，争取更多自由的代价似乎不成比例地落在大部分科普特基督徒少数族群身上。
2	Interventions planned for 2003 and beyond include improving national capacity for health and family life education, strategic planning of, and support to, adolescent and youth component in the national HIV/AIDS strategic plans and policy support for a rights approach to assisting orphans.	计划于2003年及以后采取的措施包括改进健康与家庭生活教育，在国家艾滋病毒/艾滋病战略计划中对青少年与青年问题作出战略规划，提供支助，在援助孤儿方面提供政策支助，强调权利。

Figure 1: Examples from the WMT 2019 dataset available from TensorFlow [2]. The left column is each English sentence in the dataset, and the right column is its corresponding Chinese translation. The datasets we used had either Chinese or English as the original "source" language and the other as the "target" language, but we considered all of the Chinese sentences as "source" and their corresponding English sentences as "target" for this task.

4 Methods

For our Chinese-to-English machine translation task, we designed a smaller version of the encoder-decoder model used in Google's GNMT system [5]. The model takes in an embedding of the Chinese sentence, passes this embedding to a recurrent encoder, and then passes the encoded input to another recurrent decoder that outputs logits. The output of the decoder is then fed into a scaled Luong attention mechanism that adjusts the weights after every decoder time step before the logits are converted into the translated English sentence. The original GNMT model includes 8 LSTM layers for the encoder (7 uni-directional and 1 bi-directional), and 8 RNN decoder layers (Figure 2) [8].

We trained a model with no pre-trained embedding, 1 LSTM layer for the encoder and 1 LSTM layer for the decoder as a baseline. This baseline model had 32 hidden units, a learning rate of 0.01, no dropout, and a batch size of 128. We used sparse categorical cross entropy loss and an SGD optimizer as Wu, et al did in [8]. We trained this baseline model on the same training set we used for our later models.

To decide on working hyperparameters for our Chinese-to-English dataset, we performed a hyperparameter search, training many different models for 15,000 epochs each with a batch size of 128. We tuned the learning rate between 0.01 and 1.0, the number of encoder and decoder layers between 2 layers each and 4 layers each, and the number of hidden units between 32 and 1024 on the WMT19 validation dataset. Because of resource constraints, we used smaller architectures with fewer encoder and decoder layers and no residual layers, a relatively small batch size of 128, and uni-directional instead of bi-directional layers. We used sparse categorical cross-entropy loss, a dropout of 0.2 for each layer, an SGD optimizer, and all LSTM layers. The beam width we used remained 10.

Although we performed this hyperparameter search, none of the models we initially tried converged. To improve our results, we tried using a pre-trained embedding and using a word segmenter instead of splitting by character that were menioned in Section 3. Once we started pre-processing our Chinese text data with the word segmenter and used the pre-trained FastText embeddings, our model was able to converge and our performance greatly improved. Our final model with pre-trained embeddings relied on the IWSLT parameter set given by the NMT tutorial by Luong et al and was trained for 100,000 iterations [6]. This model had 512 hidden units, a dropout of 0.2, a batch size of 128, a beam width of 10, bidirectional encoder type, 2 LSTM encoder and 2 LSTM decoder layers, and a learning rate of 1.0.

To test different parameters with the embedding, we also performed a more restricted hyperparameter search using learning rates of 0.01, 0.1, 0.5 and 2.0. All of these tests were performed using a model with 1 layer each for encoder and decoder, 64-unit embeddings, 32 hidden units, and a batch size of 32. These models were trained for 50,000 iterations each. We expected to train a larger model on more hidden units after achieving convergence of train loss on these smaller models. The difference in performance for different learning rates can be seen in the Results section.

5 Experiments/Results/Discussion

We have adapted code from a Tensorflow tutorial using an existing LSTM with attention architecture on NMT to perform Chinese-to-English translation. We have processed our subdataset of Chinese to English news text data in the same format to feed into the model, and we have confirmed that we can train models for at least 100,000 epochs on an AWS machine with 1 GPU, using a batch size of 32.

One common metric for evaluating translation quality is the bilingual evaluation understudy (BLEU) score. This score works by comparing the output of the machine translation algorithm to a set of professional human-translated sentences. Scores range from 0 to 1, where a score of 1 indicates exact match to the ground truth translation. While few high-quality translations will achieve a score of exactly 1, increased BLEU score is still indicative of translation quality. We will use the BLEU score on test data as our primary metric, and we will also look at training loss over time to evaluate model fit.

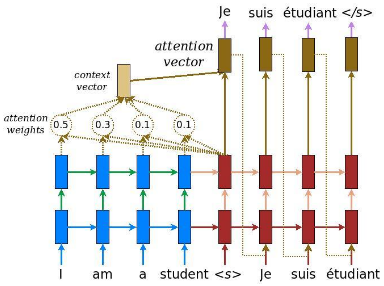
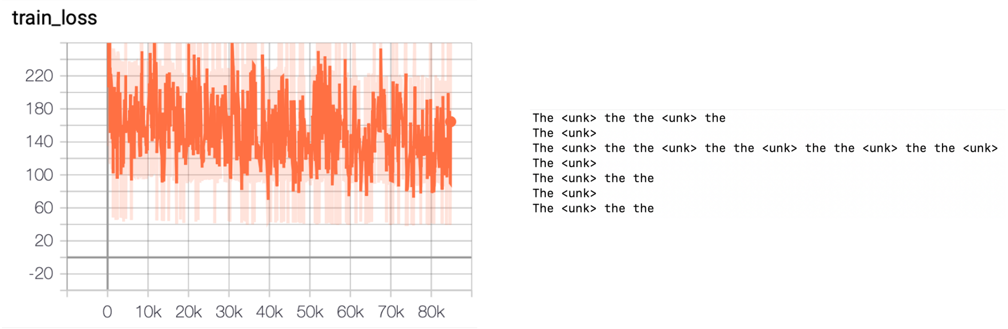


Figure 2: The model architecture of a neural machine translation (NMT) system with encoder (blue) and decoder (red) networks and an attention module (brown) [8]. This figure is from the TensorFlow tutorial we are using [6].

All of the models we tried without using a pre-trained embedding were unable to converge, despite using multiple hyperparameter sets with varying learning rates, batch sizes, number of hidden units, and number of layers. The training loss curve of the baseline model that we trained without attention is shown in Figure 3a as an example—— it was highly oscillatory and did not seem to decrease significantly over time. The maximum BLEU score we obtained from all of these models was 0.0, and the output sentence translations were fairly nonsensical (Figure 3b).



```
The <unk> the the <unk> the
The <unk>
The <unk> the the <unk> the the <unk> the the <unk> the the <unk>
The <unk>
The <unk> the the
The <unk>
The <unk> the the
```

Figure 3: (a) Training loss curve for baseline model without a pre-trained embedding or Chinese word segmentation. Loss is plotted on the y-axis and number of iterations on the x-axis. The training loss does not converge and is highly oscillatory, indicating that the fit is not good for our task. (b) Output sentence examples from the baseline model. These sentences are repetitive, low-quality translations that use only a few tokens from the vocabulary. The maximum BLEU score for this model was 0.0.

We performed an experiment on the models with the pre-trained 64-unit embeddings by testing learning rates of 0.01, 0.1, 0.5, and 2.0. However, despite training for 50,000 iterations each, it seemed that these models also did not converge despite also using the embedding (Figure 4). In fact, the model with learning rate 2.0 had a rapidly increasing loss over time. The maximum BLEU score achieved by these models was 0.063, which was not much of an improvement over our initial model, and much lower than our final model. This could have been due to the short training time in comparison to our final model, but it is more likely due to the small size and low complexity of each of the experimental models. We tried to improve performance by increasing the number of layers from 1 each to 2 each in our final model, as well as increasing the number of hidden units from 32 to 512.

Our final model with the IWSLT parameters and a pre-trained embedding performed much better, although the output translations were still not human-quality[6]. The training loss curve seemed to converge at around 80,000 iterations, and the maximum BLEU score achieved by the model on the development set was 0.2474 (Figure 5a). Although this performance is certainly an improvement over our baseline model, the translations produced by the final model seemed unrelated to the source Chinese sentences. The English translated output had repetitive beginnings (often starting with some form of "The US"), and focused on United States politics and economics even if the source sentence

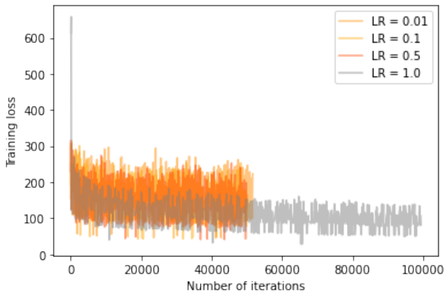
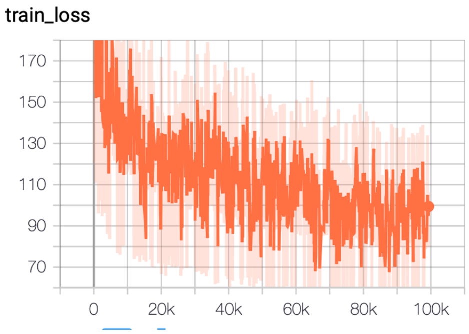
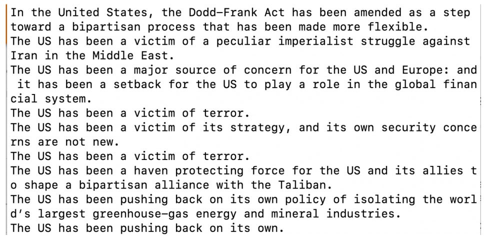


Figure 4: Learning rate experiment on models with pre-trained 64-unit embeddings, with 1 layer each for the encoder and decoder and 32 hidden units. Models were trained for 50,000 iterations. We tried learning rates of 0.01, 0.1, 0.5, and 2.0 (in orange, data not shown for the model with 2.0 learning rate). The training loss does not improve much over time for these smaller, and the final BLEU score for each of these models was very low compared to the score for our final model (gray). The training loss of the final model decreased faster, oscillated at a lower amplitude, and converged to a lower value than the other, less complex models.

had different content (Figure 5b). This may indicate that our model is biased towards political text and sentences involving the United States, perhaps due to unrepresentative training data.





```
In the United States, the Dodd-Frank Act has been amended as a step
toward a bipartisan process that has been made more flexible.
The US has been a victim of a peculiar imperialist struggle against
Iran in the Middle East.
The US has been a major source of concern for the US and Europe: and
 it has been a setback for the US to play a role in the global finan
cial system.
The US has been a victim of terror.
The US has been a victim of its strategy, and its own security conce
rns are not new.
The US has been a victim of terror.
The US has been a haven protecting force for the US and its allies t
o shape a bipartisan alliance with the Taliban.
The US has been pushing back on its own policy of isolating the worl
d's largest greenhouse-gas energy and mineral industries.
The US has been pushing back on its own.
```

Figure 5: (a) Training loss curve for the final model with a pre-trained embedding or Chinese word segmentation. Training loss is plotted on the y-axis and number of iterations on the x-axis. The training loss is still oscillatory, but decreases over time until eventual convergence after around 80,000 iterations. (b) Output sentence examples from the baseline model. These sentences are much closer to real English sentences, but they seem to be based on a common theme (United States politics) that is largely unrelated to the source Chinese sentences. The best BLEU score for this model was 0.247.

6 Conclusions and Future Work

In this work, we have obtained a dataset for a Chinese-to-English machine translation task, split it, and preprocessed the data. We trained a simple baseline model with a 1 layer LSTM encoder and a 1 layer LSTM decoder layer on our dataset, and we have trained a series of more complex models with pre-trained embeddings an various hyperparameters to improve performance. Although our final tuned model had some gaps in performance, there are other hyperparameters we can modify for better results. Some strategies for improvement would be to add more encoder and decoder layers, increase the number of hidden nodes, increase the size of the embedding vectors, further tune the learning rate, tune the dropout probability, and try different optimization algorithms. To improve the model performance further, we can incorporate even more data from the WMT19 dataset and train the model for a longer time. We could also consider using a different evaluation metric that is more applicable for sentence-level data, such as the GLEU score applied in [8].

7 Contributions

Cynthia Hao performed literature review and defined project scope; located appropriate datasets and existing model architectures for the translation task; set up computational resources on AWS; loaded, segmented, and preprocessed the data; adapted code from two NMT tutorials for model training; trained the baseline and final models; performed hyperparameter selection; and contributed to all parts of the writeup.

Zhuoer Gu performed literature review; helped with decision-making on hyperparameter tuning and model architecture; and contributed to the final presentation and the report writeup.

We would also like to express our gratitude to TA Sherry Ruan for her advice and help in our project.

8 Code

Model code (an iPython notebook with data preprocessing code and model parameters, as well as code from the Tensorflow tutorial) can be found at the following private Github link: https://github.com/hyacynth/cs230-finalproject. Please contact us if you do not have access.

References

[1] Loic Barrault et al. "Findings of the 2019 Conference on Machine Translation (WMT19)". In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 1–61. DOI: 10.18653/v1/W19-5301. URL: https://www.aclweb.org/anthology/W19-5301.

[2] Wikimedia Foundation. *ACL 2019 Fourth Conference on Machine Translation (WMT19), Shared Task: Machine Translation of News*. URL: http://www.statmt.org/wmt19/translation-task.html.

[3] Edouard Grave et al. "Learning Word Vectors for 157 Languages". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[4] The Stanford Natural Language Processing Group. *Stanford Word Segmenter*. URL: https://nlp.stanford.edu/software/segmenter.html. (accessed: 01.09.2016).

[5] Hany Hassan et al. "Achieving Human Parity on Automatic Chinese to English News Translation". In: *CoRR* abs/1803.05567 (2018). arXiv: 1803.05567. URL: http://arxiv.org/abs/1803.05567.

[6] Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. "Neural Machine Translation (seq2seq) Tutorial". In: *https://github.com/tensorflow/nmt* (2017).

[7] Ashish Vaswani et al. "Tensor2Tensor for Neural Machine Translation". In: *CoRR* abs/1803.07416 (2018). arXiv: 1803.07416. URL: http://arxiv.org/abs/1803.07416.

[8] Yonghui Wu et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *CoRR* abs/1609.08144 (2016). arXiv: 1609.08144. URL: http://arxiv.org/abs/1609.08144.