Model Compression for Chinese-English Neural Machine Translation

Stanford CS224N Custom Project

Ryan David Cunningham
Department of Computer Science
Stanford University
rcunning@stanford.edu

Abstract

State-of-the-art neural machine translation (NMT) models require large amounts of compute and storage resources, with some of the smallest NMT models clocking in at several hundred megabytes. This large size makes it difficult to host NMT models in resource-constrained environments like edge and mobile devices, requiring that the user utilize either a stable internet connection or offline dictionary. The goal of our project is to compress a pre-trained NMT model as small as possible while minimizing reduction in translation accuracy. Using a pre-trained Chinese-to-English MarianMT model, Opus-MT, we tested several size reduction techniques and observed their impact on memory size, processing speed, and BLEU.

1 Key Information to include

- Mentor: Angelica Sun (asunyz@stanford.edu)

2 Introduction

The goal of our project is to compress a pre-trained neural machine translation (NMT) model using several candidate techniques, achieving the maximum size reduction possible while maintaining reasonable performance on BLEU.

While transformer-based language models perform well on NMT tasks, their large size makes it difficult or impossible to host in resource-constrained environments. This is problematic for NMT applications since in many cases, those who are unable to speak or interpret a foreign language may find themselves in situations with access to neither cellular data nor a human translator.

NMT-enabled edge devices have the potential to provide rapid translation services in resource-constrained and infrastructure-challenged environments. Combined with speech-to-text (STT) models, full-suite NMT edge devices could be used as intermediaries, helping to avoid misunderstandings that could lead to high-profile international incidents between English-speaking and Chinese-speaking communities. Recent examples include the mass eviction of African immigrants in Guangzhou, and an oft-repeated tendency for Chinese dancers to use of blackface in the annual Lunar New Year TV show, in an attempt to "celebrate Chinese-African ties." performant. [1] [2] [3]

We selected Chinese-English as our subject pair since Chinese is consistently ranked as one of the hardest languages for English-native speakers to learn. [4] Our hope is that by investing in language understanding between these disparate linguistic communities, we can reduce the frequency of bilateral confusion and culture shock, foster deeper interactions between

speakers, and curb a sharp rise in anti-Chinese (and more broadly, anti-Asian) hate crimes across English-speaking communities.[5]

3 Related Work

Yan et al provide the most comprehensive exploration of compression techniques for language modeling.[6] Their solution won the language modeling task of the NeurIPS 2019 MicroNet Challenge, achieving a 90x parameter efficiency gain and 36x computational efficiency gain compared to the baseline model, while satisfying a required test perplexity of 35.[7]

Cheng et al provide a survey on model compression and acceleration techniques for deep neural networks (DNNs) more broadly.[8] This paper focuses on the first group of techniques, parameter pruning and quantization.

Network pruning sparsifies networks through either iterative pruning, which prunes a weight if its absolute value is less than a given threshold (either an integer value or a percent), and global pruning, which is applied to all weights simultaneously.[9]

Finally, Shu and Nakayama introduced embedding-specific compression via codes and codebooks, by casting high-dimensional 32-bit float weights into lower-dimensional integers, without a significant loss in accuracy. They achieved embedding compression rates of up to 98.4% without sacrificing accuracy in sentiment analysis and machine translation tasks.[10]

4 Approach

We tested our baseline model against a never before-seen test set, then implemented several compression techniques for our NMT model Opus-MT, built by University of Helsinki researchers upon the MarianMT framework.

4.1 Architecture

Opus-MT utilizes a basic encoder-decoder structure with multi-head self attention. Both the encoder and decoder have 5 layers each, comprised of normalization and linear modules. This architecture is illustrated in table 2.

4.2 Baselines

Baselines were established by running inference against the WMT17 Chinese-English translation task.[11] The Opus-MT model was trained on data from the Tatoeba Challenge.[12] The WMT17 test data had to be preprocessed to extract source text, and Opus-MT was provided out-of-the-box with code from the Hugging Face Transformers library.[13]

Since PyTorch 1.8 does not yet support CUDA-based quantized models, all inference had to be run on CPU. This resulted in longer runtimes than usual, but provides an apples-to-apples comparison between the different methods.

4.3 Pruning

With the notable exception of recent sparse transformer models like the Switch Transformer[14], most state-of-the-art transformer models like GPT-3 make use of dense connectivity in their networks. There is some research demonstrating the efficacy of sparse connectivity in some applications,[15] but we are curious to see how outright elimination (rather than size reduction) of model weights and activations affect NMT performance without post-training.

We used a vanilla PyTorch implementation for L1 unstructured pruning, which identifies and removes module elements with the lowest L1-norm, the sum of all magnitudes of vectors in a given space. Instead of iteratively pruning the lowest X% of connections by layer, we built a global pruning function that prunes the lowest X% of connections across the entire model.

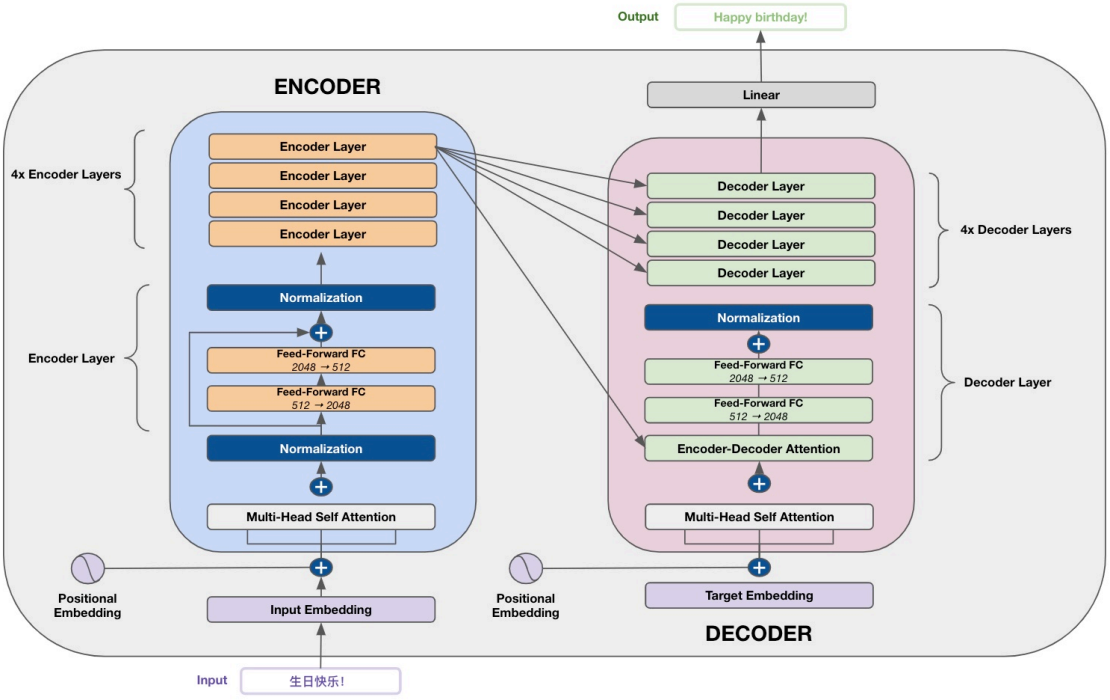

Figure 1: Our Opus-MT model structure.

4.4 Quantization

Quantization is a process that reduces the bits representing a number. Converting a value represented by float32 to int8 sacrifices precision for storage and speed, achieving up to a 4x reduction in size for the variable. This is a growing field of research within deep learning, especially for resource-constrained environments, as most models use float32 for weights and activations by default. When quantization techniques are applied tensor-wise to layers or the entire model, developers can achieve significant improvements in model compute and storage requirements.[6]

Model weights, biases, and layers can all be quantized. *Dynamic quantization* is the simplest method that compresses select model layer activations to the specific bitwidth, but not weights or biases. In this study, we strictly observe the impact of dynamic quantization on linear activations.

4.5 Embedding Compression

Embedding compression condenses high-dimensional word embeddings into lower-dimensional quantized codes. Whenever a word is assigned an independent embedding vector, the number of parameters required to store all embeddings can require a massive amount of storage space. Our original Marian model utilizes a vocabulary of 65,000 words in 512 dimensions, requiring over 33M embedding parameters (all in float 32) to store the vocabulary. While this does not impact performance, since only a few words are pushed through the model at any time, it does impact storage and memory requirements.

For our experiment, we used the PyTorch implementation of the original neuralcompressor repo, with some modifications for better generalizability.[16] Neuralcompressor represents each word w with a code C_w:

$$C_w = C_w^1, C_w^2, ..., C_w^M$$
$$Dog = [14, 6, ..., 9]$$
$$Dogs = [14, 6, ..., 10]$$

Each word contains M codes, and each subcode C_w^i is an integer number in [1, K]. Words of similar meaning are expected to have similar codes. In this way, theoretical embedding compression is achieved by reducing the embedding dimensionality from 512 in our model to M, in addition to weight quantization by converting weights to integer codes of precision K.

5 Experiments

5.1 Data

The Opus-MT model was trained on Chinese-English translation data used for the Tatoeba Translation Challenge.[12] Most of these translations resemble natural human conversations. Our test set, collected separately, consists of 2,000 Chinese-English translations from WMT17, which are Chinese-language article headlines translated into English.[17]

5.2 Evaluation Method

Our models are evaluated on file size (in memory), inference speed, and translation accuracy via BLEU-4. We use SacreBLEU to compute BLEU scores on a per-translation basis.[18]

For our embedding compression, we measured file size, theoretical compression (parameter dimensions), code length in bits, and drift via mean Euclidean distance. Code length in bits is measured by $Mlog_2K$.

5.3 Experimental Details

The base model was retrieved out-of-the-box via Hugging Face, with no modifications or fine-tuning on WMT17. Since PyTorch 1.8 does not support GPU inference for quantized models, all experiments were run on CPU for apples to apples comparison, specifically a 2.6GHz Intel Core i7-8850H CPU. We used a batch size of 50 on 1,000 translations in the test set.

5.4 Results

Our initial results demonstrated successful compression on our linear quantized model with minimal loss in BLEU. For our pruned model, PyTorch does not yet support structured pruning as usage of sparse tensors is still in beta, meaning that the storage and memory footprints do not change.

Model Compression Results			
Metric	Base	Pruned Model (2x)	Quantized (INT8)
File Size	303.9 MB	303.9 MB*	207.8 MB
Time / Translation	1,101.7ms	9,394.9ms	922.3ms
BLEU	29.56	23.12	28.68
Compression	–	–*	1.5x
Speedup	–	0.12x	1.2x
Accuracy Loss	–	(21.8%)	(3.0%)

Table 1: Since PyTorch 1.8 does not yet support sparse tensors in production, storage and memory usage remain unchanged, but 50% of all layer tensors are sparsified.

A global pruning rate of 50% successfully sparsified layers at an average of 50% per layer, however our implementation failed to avoid a significant loss in accuracy. This halted further investigations of pruning rates of 4x, 8x, and 16x as originally planned.

Dynamic quantization using did not achieve the expected compression rate of 4x, netting only 1.5x. Since vanilla PyTorch only quantizes the activations in quantize_dynamic, and not the original embeddings, this led us to conclude that greater theoretical compression rates may be achieved by focusing on the embeddings, rather than the layers.

For our embedding compression, we examined two different sets of $M\ x\ K$ coding, 32 x 16 and 64 x 8. Stored as binary files, both achieved significant reductions in memory store with minimal drift in Euclidean distance, indicating most words maintained their semantic representation despite loss of precision.

Embedding Compression Results			
Metric	Original	32x16	64x8
File Size	416.2 MB	7.4 MB	7.5 MB
Dimensionality	65,000x512 (FP32)	65,000x32 (INT16)	65,000x64 (INT8)
Code Length	2,560 bits	128 bits	192 bits
Mean Euclidean Distance	–	0.378	0.543
Storage Compression	–	56.2x	55.5x
Parameter Compression	–	32x	32x

Table 2: File size represents embeddings stored in word2vec format.

6 Analysis

6.1 Qualitative Results

Beyond investigating performance enhancements on storage, speed, and accuracy, we were interested to see if there were identifiable trends in some of the lowest performing translations. Table 3 contains a sample of the worst translations from each model.

Immediately, it is clear that our pruning method resulted in harsh repetition loops that do not seem to be reflected in the base or quantized models. We investigate this further in 6.2.3.

6.2 Impacts to Accuracy

6.2.1 Translation Length

We investigated how the accuracy of our NMT model is related to the length of the Chinese input. Since Chinese does not use whitespace to separate words, word boundaries are considered ambiguous, which provides an additional challenge when assessing the length of a source text. We use Chinese text segmentation techniques provided by the jieba package to split the string into distinct characters groupings that correspond to a given Chinese dictionary.[19]

Running jieba.cut on 这是一个激动人心的时刻 yields the following segmentation:

这是, 一个, 激动人心, 的, 时刻

'This is', 'an', 'exciting', *possessive indicator*, 'moment'

Selected Inaccurate Translations			
	Base	Pruned (2x)	Quantized (INT8)
Source	**然而，即使是按照普京先生一向内敛的标准来看，普京也只是浅浅微笑，几乎未流露出热情之意。**	**拜尔斯因此跻身像迈克尔-菲尔普斯（Michael Phelps）一样的“几十年一遇运动员”的行列，他们将各自领域的体育项目提升到新高度**	**据《明镜周刊》的报道，酒店很快对所有指控予以澄清，声称一切都是“疏忽”并且向朗兹曼表示“歉意”。**
Reference	But Mr Putin's smile looked thin and he was hardly oozing warmth even by his own restrained standards.	The achievement puts her in the same league as once-in-a-generation athletes like Michael Phelps who have taken their sports to new heights.	The hotel itself was quick to brush off any accusations, calling what happened an "oversight" and "apologizing" to Lanzmann, Der Spiegel reports.
Target	However, even in the light of the standard that Mr. Putin has always held, Putin is merely a light smile with little enthusiasm.	By the way, by the way,【...】by the way, in the way of the Michael-Phelps, by the way, by the way, by the way, 【...】the way, of the way, that of the sports in the various fields of which they have been, 【...】, the way, the way, 【...】the things of the way, the way, 【...】	According to the Sunglasses Weekly, the hotel quickly clarified all the charges, claiming that everything was “neglect” and expressing “apologies” to Lunzman.
BLEU	1.963	0.243	2.533

Table 3: 【...】represent repetitive n-grams that have been omitted for readability.

We then bucketed jieba segment lengths into bins of 5 words each to reduce volatility in regression plots.

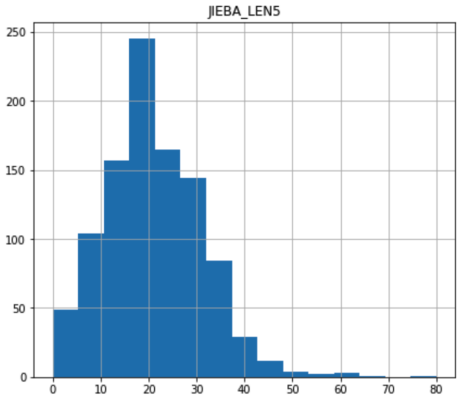

Figure 2: Histogram of jieba word segments for all source tests in test set.

However, none of the three models exhibited any statistically significant reduction in translation accuracy with longer source texts. While the data distribution resembles a polynomial relationship, confidence intervals are too wide to draw a statistically significant conclusion.

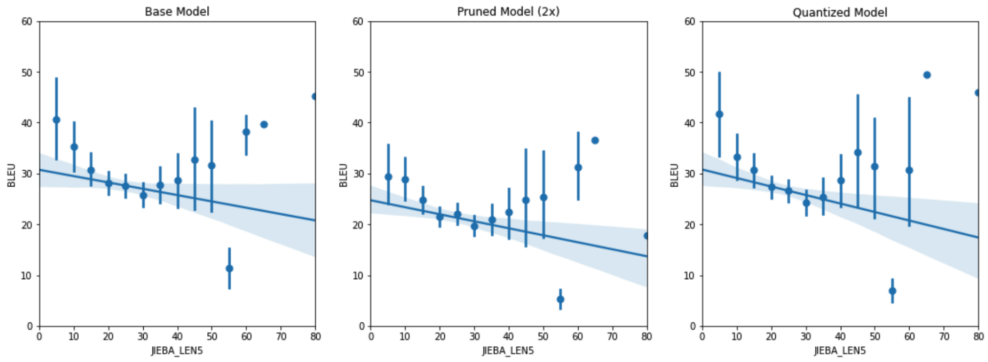

Figure 3: Linear regression plots for jieba word segmentation bins vs. BLEU-4 score.

6.2.2 Entity Recognition

The next item we wanted to investigate was the frequency of named entities in the source text. Our qualitative assessment highlighted several bad translation examples which contained named entities cast as strange or literal translations ("Der Spiegel" → "Sunglasses Weekly").

We used spacy to identify and tally named entities in our English reference translations.[20] We then plotted a regression chart to examine the impact of entity count on translation accuracy. However, no statistically significant relationship exists between the number of detected entities and translation accuracy.

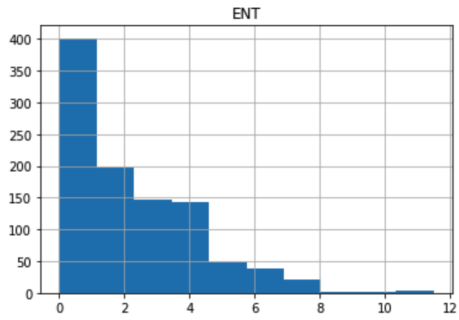

Figure 4: Histogram for number of entities recognized in test set reference translations.

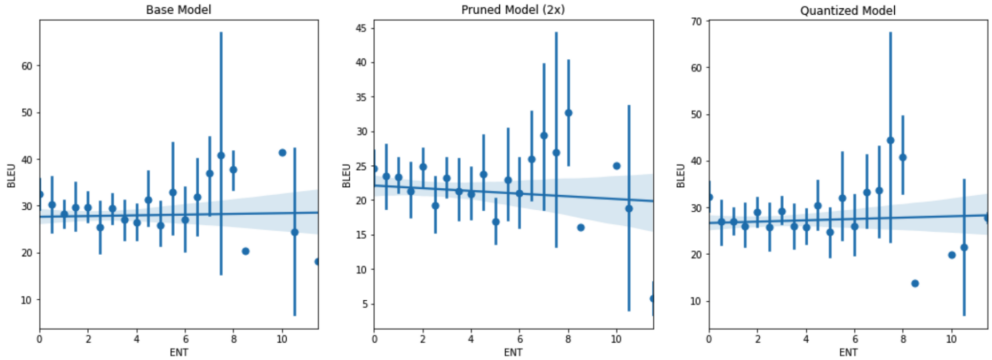

Figure 5: Histogram for number of entities recognized in human-reference translations in test set.

6.2.3 Repetition

Repetitive utterances in translated texts may be a function of inefficient or inelegant model compression techniques. Notably, our worst performing translations from the pruned model had oft-repeated ngrams which were not present in the base or quantized translations for the same text.

To examine this, we extracted all 1, 2, 3, and 4-grams from the reference translations, calculated their frequency distribution, then determined the mean n-gram frequency for that translation. We then repeated the process for our target translations. The delta between reference and target mean n-gram frequencies represents our repetition score.

Our pruned model had a significantly higher repetition score than either the base or quantized model.

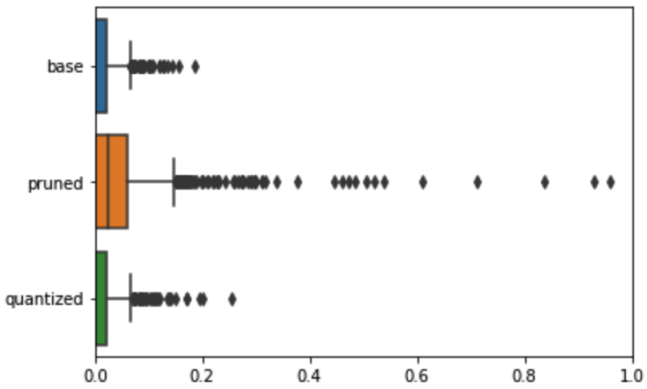

Figure 6: Boxplot for repetition score (target ngram repetition - reference ngram repetition).

Compared to either model, the variance in the pruned model was incredibly wide, indicating a significant loss in translation consistency.

7 Conclusion and Future Work

Our intent was to explore two methods of model compression for NMT-specific applications. Given the high compression rates achieved via embedding compression, we believe a successful implementation of embedding compression and linear quantization should comprise the majority of NMT-specific model compression gains.

While our quantization efforts were limited to 8 bit integers, we would like to explore 4 bit and as low as 2 bit storage for each subcode in our embedding codebook. Based on the performance of the linear quantization model, we expect to implement this with minimal loss to accuracy.

Conversely, pruning efforts proved that while NMT models can sacrifice precision without drastically impacting accuracy (even without post-training), outright sparsification without post-training has severe impacts to accuracy. In short, weights cannot simply be deactivated without consequence.

Finally, we intend to explore exporting our compressed model to edge-friendly formats via ONNX and deepC, [21] [22] to deploy alongside a Chinese STT model. [23] Ultimately, if deployed on an edge device with an integrated microphone, this combination can create a low-cost portable Chinese-English translator without the need for a stable internet connection.

8 Acknowledgements

This paper and the underlying research would not have been possible without the tremendous support of my mentor, Angelica Sun. Her enthusiasm for the budding field of quanti-

zation in deep learning was invaluable in determining the initial, intermediate, and ultimate scope of this study, and her willingness to independently attempt her own implementations in parallel helped verify blockers in the current PyTorch quantization framework for transformers that could not be resolved without severe source code modification. This provided rapid feedback which unblocked my own investigations.

I am also grateful for the insightful comments provided by other peers and research staff, including Elissa Li, Lauren Zhu, William Ellsworth, and Sergio Charles, many of whom sacrificed Fridays and Saturdays to brainstorm and provide guidance. My sustained enthusiasm for deep learning and model compression is due in no small part to the generosity of all those mentioned.

References

[1] Danny Vincent. Africans in china: We face coronavirus discrimination - bbc news. https://www.bbc.com/news/world-africa-52309414, 4 2020.

[2] Lunar new year: Chinese tv gala includes 'racist blackface' sketch - bbc news. https://www.bbc.com/news/world-asia-china-43081218, 2 2018.

[3] Echo Xie. China' s lunar new year tv extravaganza hit again by blackface scandal | south china morning post. https://www.scmp.com/news/china/diplomacy/article/3121588/chinas-lunar-new-year-tv-extravaganza-once-again-tainted, 2 2021.

[4] Dylan Lyons. The 6 hardest languages for english speakers to learn. https://www.babbel.com/en/magazine/6-hardest-languages-for-english-speakers-to-learn, 2 2021.

[5] The rise in anti-asian attacks during the covid-19 pandemic : 1a : Npr. https://www.npr.org/2021/03/10/975722882/the-rise-of-anti-asian-attacks-during-the-covid-19-pandemic, 3 2021.

[6] Zhongxia Yan, Hanrui Wang, Demi Guo, and Song Han. Micronet for efficient language modeling, 2020.

[7] Announcements | micronet challenge. https://micronet-challenge.github.io/, 2019.

[8] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks, 2020.

[9] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks, 2015.

[10] Raphael Shu and Hideki Nakayama. Compressing word embeddings via deep compositional code learning, 2017.

[11] 2017 second conference on machine translation (wmt17). http://www.statmt.org/wmt17/, 9 2017.

[12] Tatoeba-challenge/readme.md at master ·helsinki-nlp/tatoeba-challenge ·github. https://github.com/Helsinki-NLP/Tatoeba-Challenge/blob/master/models/zho-eng/README.md.

[13] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[14] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021.

[15] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019.

[16] Raphael Shu and Hideki Nakayama. Compressing word embeddings via deep compositional code learning. In *International Conference on Learning Representations (ICLR)*, 2018.

[17] Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[18] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics.

[19] Github - fxsjy/jieba: 结巴中文分词. `https://github.com/fxsjy/jieba`.

[20] Github - explosion/spacy: industrial-strength natural language processing (nlp) in python. `https://github.com/explosion/spaCy`.

[21] Junjie Bai, Fang Lu, Ke Zhang, et al. Onnx: Open neural network exchange. `https://github.com/onnx/onnx`, 2019.

[22] Github - ai-techsystems/deepc: vendor independent deep learning library, compiler and inference framework microcomputers and micro-controllers. `https://github.com/ai-techsystems/deepC`.

[23] Github - kaituoxu/speech-transformer: A pytorch implementation of speech transformer, an end-to-end asr with transformer network on mandarin chinese. `https://github.com/kaituoxu/Speech-Transformer`.