Cross-cultural communication using Machine Translation

MD ABU QUWSAR OHI and SUJESH PADHI

CCS Concepts: • **Computing methodologies** → **Machine translation**.

Additional Key Words and Phrases: machine translation, natural language processing, deep learning

1 INTRODUCTION

1.1 Project Outline

Human Language is one of the most natural forms of signals that ages beyond centuries. For centuries, human beings have tried to translate one language to another to communicate across the world. Language translation is the process of converting information given in one language to another. It has become an important area of focus for daily life communication with cross-lingual individuals. Machine translation (MT) conveys a similar action that is accomplished using machines/computers to assist humans. This is achieved by training machine learning models with language data and using them for automated language translation.

MT finds utility in various areas due to its multiple advantages over human translation such as providing faster and more consistent translation with wider cross-lingual capabilities. This also encourages users from non-widely spoken languages to collaborate efficiently. Furthermore, an accurate machine translation system can reduce the risk of translation errors compared to human translators. There is an added advantage of flexibility and availability using any machine translation model; they provide easy deployability, cost-effectiveness, and a reduced error rate. All these exceptional advantages motivated researchers to investigate the improvement of MT systems, and eventually it has gained importance over the years.

Due to the robustness of neural networks, neural machine translation (NMT) has been gaining popularity in recent years. As a result, most of the current MT systems are built upon deep learning strategies. Although the current NMT strategies show high accuracy, they also have a large number of parameters making them computationally complex, difficult to train, and requiring expensive infrastructures [3].

In this project, we used the Text-to-Text Transfer Transformer (T5), which is a transformer-based encoder-decoder model which has been pre-trained on a large dataset named Colossal Clean Crawled Corpus (C4) [17]. We employed layer freezing techniques to train the model further in two rounds using the *IWSLT 2017* dataset [4]. The dataset was divided into training and testing sets using the T5 internal libraries before the training process. After training was complete and the validation scores are obtained, we tested the model using the testing dataset and compared the testing scores to the validation scores to assess the performance of the model. We used the loss value and Bilingual Evaluation Understudy (BLEU) score to evaluate the performance. And our model gave a decent performance in translating multiple languages, including both English and non-English language mappings.

1.2 Related Work

Relevant works in the area of machine translation have been challenging due to the use of textual representations which contain a sequential structure, wherein different orders or vocabularies can change the context of a sentence. Therefore, a variety of machine translation models have been explored, such as the rule-based models [9], the statistical models [12], and the neural models [1]. Additionally, there exist hybrid models which combine multiple strategies into a single translation model [8].

Due to the sequential pattern, recurrent neural network (RNN) architectures were implemented for NMT [2]. Often, these RNN architectures implement an encoder-decoder scheme, where the encoder converts the input language into a context vector which the decoder further decodes into the target language [2]. Most encoder-decoder schemes are implemented using sequence-to-sequence modeling [16], where the encoder outputs a hidden state by processing the input. The hidden state is then passed to the decoder which decodes the information in the target language.

Additionally, RNNs suffer from the vanishing gradient problem and, therefore cannot attain better performance for larger sequences. To store the past information more accurately and to mitigate the vanishing gradient, different recurrent architectures such as long short-term memory (LSTM) [10], gated neural network (GRU) [5] have been introduced. However, LSTM tends to work better than other recurrent neural networks, specifically for language translation tasks [5].

Although recurrent architectures performed better than most deep learning architectures and could explore bi-directional relationships [15], they could not explore complex one-to-many and many-to-many relationships in a text. To mitigate this issue, an attention network was introduced that can explore complex language relationships. Moreover, transformer architectures are more amenable to parallel processing while considering the contextual connectivity of all previous inputs. They have also been investigated in machine translation [18]. Transformer architectures have been further investigated for deeper transformer architectures [19], highlighting deeper and highly parameterized transformer architectures that perform better than shallow and less-parameterized models. As training highly parameterized architectures is challenging, Prato et al. introduced a quantization method specifically designed for transformer-based machine translation models [13]. Yao et al. further investigated multi-modal machine translation, which received an image along with a text that needs to be translated into a different language [21]. Raganato et al. [14] modified the positional encoding of the encoder so that the transformer could attain better features. Meta further implemented a highly parameterized transformer-based machine translation model that can work on 202 languages [7].

Moreover, numerous research works have been done on our selected dataset (IWSLT 2017) for machine translation. Correia et al. [6] introduced a modified attention function to provide sparsity on the attention matrix. Want et al. [20] introduced a parameter differentiation-based method that uses transformer architecture. The authors implemented many-to-many translation mappings using their model, similar to our work. Zeng et al. [22] improved the translation capability of transformers by adding lexical constraints to the model and reported one-to-one translation scores on the IWSLT 2017 dataset.

2 MATERIALS AND METHODS

2.1 Neural Network Information

Based on the previous research, that has been conducted we decided to use the T5 transformer-based encoder-decoder model. It is a pre-trained model that has been trained and tested for a diverse set of tasks, including language modeling, translation, summarization, and question-answering. It was trained using an unsupervised strategy with a maximum of 100 special tokens, followed by a supervised training strategy. In the supervised training strategy, the model was trained with a task prefix structure: "{task_prefix}: {sentence}", where the task prefix highlighted the task the model had to perform. As we are aware that the model was pre-trained on the large C4 dataset, making it well-suited to be used out of the box. Since our objective was language translation and the T5 model comes with in-built mappings for English-to-German and French-to-Romanian, it was a suitable choice. Fig. 1 shows some of the out-of-the-box capabilities of the model.

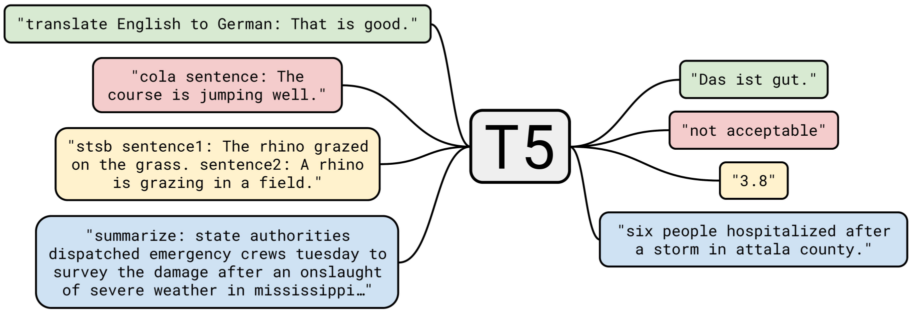

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Fig. 1. Capabilities of the T5 model.

Further, we generated a data pipeline that can be configured with a language-specific mapping for the translation. This language-specific mapping is the task prefix that we discussed earlier and that contains information on the 'FROM' and 'TO' languages of the translation, which states the information for the required language translation. Using this approach, we were able to train only a single model for all of the language-translation maps. During fine-tuning, we used the default tokenizer of the T5 model, which uses SentencePiece sub-word token sequences.

2.2 Datasets

Identifying a suitable dataset plays an essential role in the training of any machine learning model. As mentioned earlier, we used the IWSLT 2017 dataset, which contains six language mappings given in Table 1. Therefore, we used all of the language mappings to train the model.

As T5 is a multi-task model, it receives a specific pattern of input: *"task: description"*. Here, the task defines the action the model has to perform, followed by a description that serves as the actual input for the mentioned task. As we are training the model for language translation, we use the following input pattern:

"translate X to Y: sentence-to-translate"

Where "X" is the source language in which the "sentence-to-translate" is written. The objective of the model is to convert the "sentence-to-translate" from the source language "X" to the destination language "Y".

		TO					
		English	**German**	**Italian**	**Dutch**	**Romanian**	**French**
FROM	**English**	NA	X	X	X	X	X
	German	X					
	Italian	X			X	X	
	Dutch	X		X		X	
	Romanian	X		X	X		
	French	X					

Table 1. The translation mappings of the dataset are given in the table. Cross (X) sign indicates that the language (given in the row) has an existing translation mapped to the other language (given in the column).

2.3 Training and Fine-tuning the Network

The training involved a series of two complex rounds of activity. In order to understand the training process, we need to understand that the T5 encoder and decoder where each has six blocks of neural networks, and each of them is constructed using an attention mechanism; thus we have twelve blocks of neural networks. The model comes with *60.5 million* parameters, and so we used *16-bit precision* numbers in the training pipeline.

In the first round of training, we froze the first ten blocks and left the last two blocks of the decoder aside, training with a learning rate of 10^{-4}. This was done to preserve the weights of the pre-trained T5 model and only fine-tune the last two blocks. In the second round, those ten blocks were unfrozen and trained together with the newly tuned weights in the last 2 blocks. The second round of training involved cumulative learning rates, i.e., a learning rate of 10^{-6} from the first ten blocks and a learning rate of 10^{-4} from the newly acquired weights, thus preparing the final model ready for language translation.

We used stochastic weight averaging (SWA) for better generalization [11]. It is a weight optimization mechanism technique. SWA takes this optimization process by averaging the multiple points along the trajectory of the optimizer with a cyclical or constant learning rate. This averaging technique leads to wider optima and improved generalization.

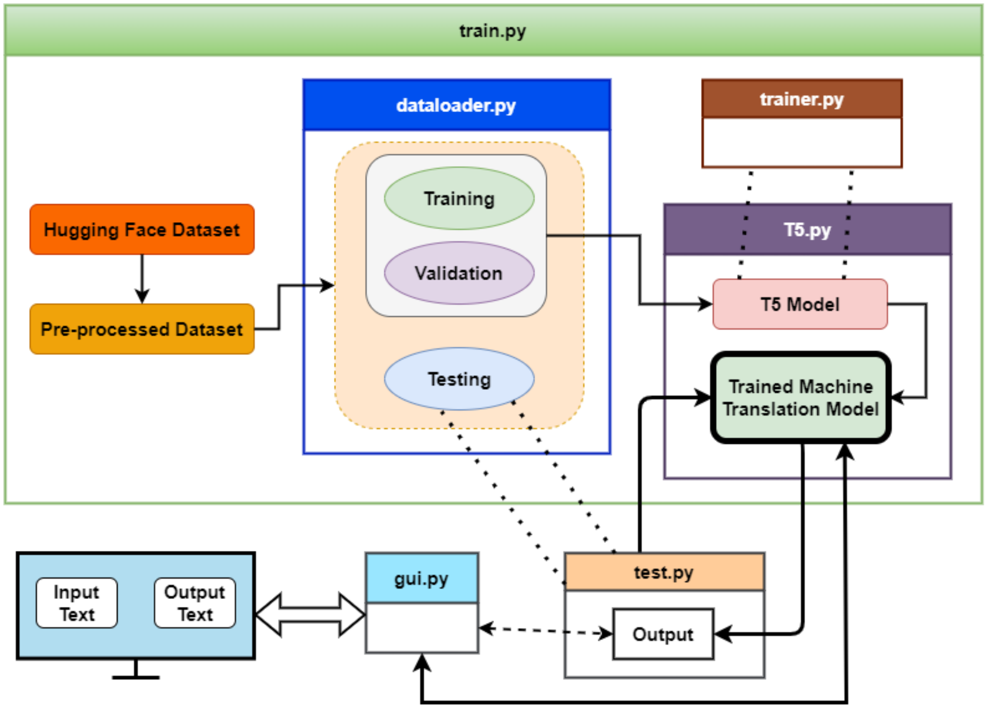


Fig. 2. Block Diagram of the Machine Translation System

A complete view of the project pipeline can be represented using the block diagram in Fig. 2. The model was trained for 20 epochs with a batch size of 40 initially, yielding a loss of 0.56 and 0.50 during with the validation and testing datasets respectively. A similar relationship was observed for the other performance metrics named BLEU score. It showed 0.21 and 0.25 for validation and testing respectively as shown in figure 3 for the English-Italian translation. It is important to note that these BLEU scores are close to the actual benchmark scores observed using other S-O-T-A models. We also further trained the model on a server with an Intel i9 processor with 64 Gigabytes of system memory (RAM) and with the NVIDIA GeForce RTX 3090 with 26 Gigabytes of GPU memory for a duration of 24 hours. Finally, we used the grad.io[1] to create a graphical user interface that connects to the back end of the model. The complete project has been developed and is available on the public GitHub repository[2].

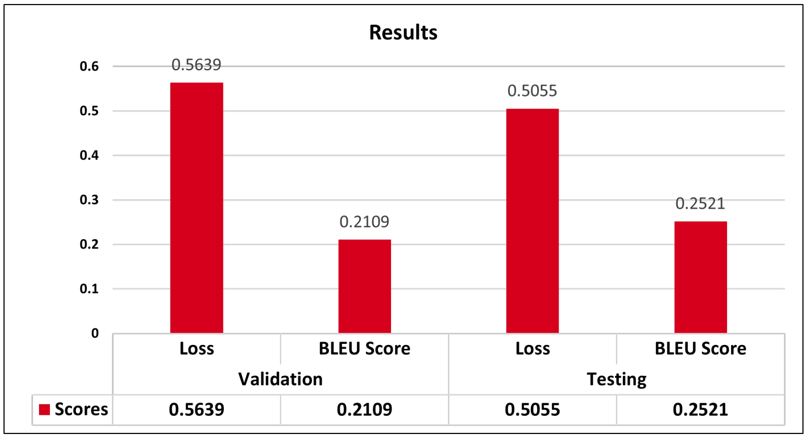


Fig. 3. Bar chart representing initial performance scores after 20 epochs of training

3 RESULTS

After 24 hours of training the model, it used the internal libraries to do validation with the respective dataset and generated two performance metrics - the loss value and the BLEU score. Both the scores are represented using values ranging from 0 to 1. These two metrics were used for evaluating the model's performance for each language mapping. In order to yield the evaluation metrics, we turned on the inference mode of the model and provided the language details as per the input pattern. The input values correspond to the ids present in the dataloader.py file for each language mapping, and we have the flexibility to evaluate the performance metrics of all the individual language mappings. The score table 2 shows the score captured for the loss and BLEU for all the individual language mappings under consideration in this project.

Fig. 4 shows the translation from English to Italian and the back-translation using google translate. Similarly, we could observe Italian to French and its corresponding back-translation to English for correctness in Fig. 5.

4 DISCUSSION AND CONCLUSION

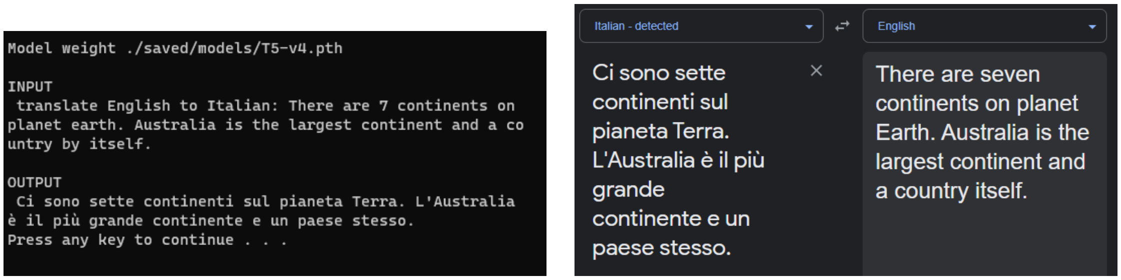
4.1 Result Analysis

As observed in Table 2 the loss in validation versus testing represents that they are close to each other, and that justifies the good performance of the model. The closeness indicates that the model's performance was good with unseen data.

[1] Gradio website: https://gradio.app/
[2] GitHub project repository: https://github.com/QuwsarOhi/NMT-project

Language Mapping		Validation		Testing	
From	To	BLEU Score	Loss	BLEU Score	Loss
English Mapping					
English	Italian	0.196	0.438	0.195	0.420
Italian	English	0.247	0.737	0.250	0.673
English	German	0.183	0.670	0.187	0.703
German	English	0.263	0.644	0.254	0.680
English	Dutch	0.152	0.538	0.210	0.415
Dutch	English	0.202	0.909	0.273	0.683
English	Romanian	0.171	0.613	0.192	0.589
Romanian	English	0.279	0.632	0.277	0.666
English	French	0.210	0.574	0.290	0.472
French	English	0.230	0.764	0.305	0.605
Non-English Mapping					
Italian	Dutch	0.103	0.619	0.130	0.515
Dutch	Italian	0.108	0.642	0.126	0.544
Romanian	Dutch	0.115	0.574	0.137	0.507
Dutch	Romanian	0.092	0.894	0.121	0.746
Romanian	Italian	0.146	0.499	0.154	0.513
Italian	Romanian	0.115	0.788	0.132	0.732

Table 2. Loss and BLEU score for all the language mappings



```
Model weight ./saved/models/T5-v4.pth

INPUT
 translate English to Italian: There are 7 continents on
planet earth. Australia is the largest continent and a co
untry by itself.

OUTPUT
 Ci sono sette continenti sul pianeta Terra. L'Australia
è il più grande continente e un paese stesso.
Press any key to continue . . .
```

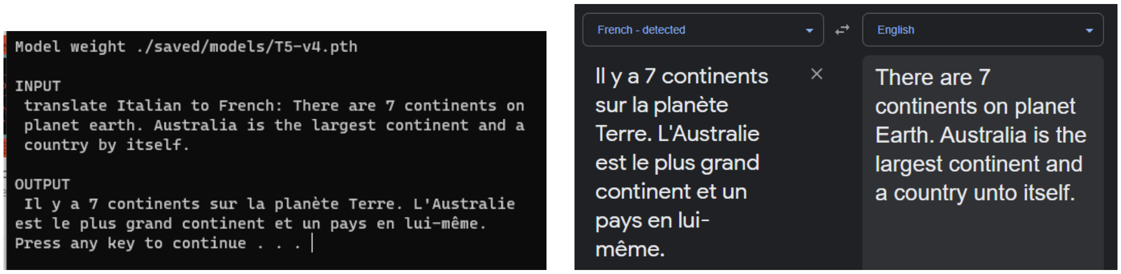
Ci sono sette continenti sul pianeta Terra. L'Australia è il più grande continente e un paese stesso.

There are seven continents on planet Earth. Australia is the largest continent and a country itself.

(a) English to Italian translation using our trained model

(b) Italian to English back-translation using Google Translate

Fig. 4. Snapshot of English mapping translation and back-translation



```
Model weight ./saved/models/T5-v4.pth

INPUT
 translate Italian to French: There are 7 continents on
 planet earth. Australia is the largest continent and a
 country by itself.

OUTPUT
 Il y a 7 continents sur la planète Terre. L'Australie
est le plus grand continent et un pays en lui-même.
Press any key to continue . . .
```

Il y a 7 continents sur la planète Terre. L'Australie est le plus grand continent et un pays en lui-même.

There are 7 continents on planet Earth. Australia is the largest continent and a country unto itself.

(a) Italian to French translation using our trained model

(b) French to English back-translation using Google Translate

Fig. 5. Snapshot of non-English mapping translation and back-translation

Similarly, the BLEU scores show that the model was able to perform better translation. The closeness of the scores proves that the model was not over-fitted and was able to perform well with the unseen data. Further in Fig. 4 and 5 we did an English-to-Italian and Italian-to-French translation using the model in inference mode and obtained the output correctly. To evaluate the model output, we used Google Translate as a means of back-translation. This Google Translate output was similar to the original input. Moreover, in the case of the Italian-to-French translation, we provided the input in English even though the instructions were Italian-to-French, and we observed the model detect Italian and do an accurate translation that could be verified by a back-translation to English using Google Translate. This gives us the view that the model was trained well to address English and non-English language translation and performed well with accurate and precise translations.

4.2 Results: Comparison to previous work

Correia et al. introduced [6] used their model to train English to German language translation and achieved a BLEU score of 0.269, whereas our score is 0.187. Compared to the previous research work [6], our method achieved competitive performance, as we built our model for many-to-many translation mappings. Whereas the authors [6] implemented a one-to-one translation mapping for the model.

Want et al. [20] used vanilla transformer architecture with increased parameters, having 1.25 times more parameters than our model. The authors reported that they got a 0.265 BLEU score on the English-to-Romanian translation, whereas, for a similar translation, our method got a score of 0.192. For non-English mapping, the authors reported 0.235, whereas our method scored 0.154 for Romanian to Italian language translation. Zeng et al. [22] reported a one-to-one translation from English to German of 0.342 BLEU score. Thus comparatively, our model attained a BLEU score of 0.187.

Our model does not perform similarly compared to S-O-T-A architectures due to two reasons: a) a low number of parameters, and b) many-to-many language translation. As we are focusing on the many-to-many translation, the model has to memorize more information compared to the one-to-one language translation. In contrast to other works, we did not use a highly parameterized model for our purpose. Hence, it limited the performance of the model.

4.3 Results: Importance and Limitations

On the contrary, the findings give us a better view of a machine translation system that can be used to perform language translation faster and more accurately. This can be used in real-world scenarios, and we have developed a GUI that could be accessed by running the gui.py file. It generates an output with a URL[3] that could be accessed to view the interface. The interface takes a text as input, and on the selection of the destined language, it can generate the output. It gives a better view of the feedback we received for the implementation of a real-time translation system.

One limitation, as well as an advantage of the model, it's the ability to detect the source language automatically before translation without the need to provide the language for the text to be translated. Alternatively, the model takes approximately less than 5 seconds to generate output, which is slower when compared to real-time usage as the expectation would be in milliseconds. Additionally, the translation is sequential, and if the application is deployed in a production setup then we might be using parallel processing, so the performance of the model might be slower due to higher latency. The model doesn't perform well with the increase in the text size beyond 100 words, so it requires more training and hyper-parameter tuning for performance optimization to get better and more accurate results. We also need to consider a wide variety of languages to broaden the scope of the model for a better fit in the real world.

[3] Localhost site: http://127.0.0.1:7860

4.4 Conclusion and future scope

This project concluded in the development of a multi-language translator using machine translation successfully with an interactive GUI that could be used efficiently for generic translations regularly in our daily life. Future research in this area includes the identification of other widely spoken as well as rarely spoken languages to facilitate more diversification in cross-cultural communication. Training the model on ancient languages could also make it useful for Linguistic and Philology. Considering nations like Indonesia, Nigeria, India, and many more, speak more than 300 different languages and so research on the implementation of a machine translation system on a wearable device could help in better collaboration among people from different backgrounds.

REFERENCES

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

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

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the iwslt 2017 evaluation campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation*. 2–14.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[6] Gonçalo M Correia, Vlad Niculae, and André FT Martins. 2019. Adaptively sparse transformers. *arXiv preprint arXiv:1909.00015* (2019).

[7] Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672* (2022).

[8] Marta R Costa-Jussa and José AR Fonollosa. 2015. Latest trends in hybrid machine translation and its applications. *Computer Speech & Language* 32, 1 (2015), 3–10.

[9] Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation* 25, 2 (2011), 127–144.

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

[11] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2019. Averaging Weights Leads to Wider Optima and Better Generalization. arXiv:1803.05407 [cs.LG]

[12] Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys (CSUR)* 40, 3 (2008), 1–49.

[13] Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. 2019. Fully quantized transformer for machine translation. *arXiv preprint arXiv:1910.10485* (2019).

[14] Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2020. Fixed encoder self-attention patterns in transformer-based machine translation. *arXiv preprint arXiv:2002.10260* (2020).

[15] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45, 11 (1997), 2673–2681.

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

[17] Tensorflow. 2008. C4 Tensorflow Datasets. https://www.tensorflow.org/datasets/catalog/c4. [Online; accessed 13-April-2023].

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

[19] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787* (2019).

[20] Qian Wang and Jiajun Zhang. 2022. Parameter differentiation based multilingual neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11440–11448.

[21] Shaowei Yao and Xiaojun Wan. 2020. Multimodal transformer for multimodal machine translation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 4346–4350.

[22] Weiyuan Zeng and Cong Liu. 2021. Improving Lexical-Constraint-Aware Machine Translation by Factoring Encoders. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–6.