# Machine Translation with Transformer

**Zhenyu Guan**
2100017795
Peking University
2100017795@stu.pku.edu.cn

**Yixiang Liu**
2200010875
Peking University
2200010875@stu.pku.edu.cn

## Abstract

In this paper, we built upon the original English-French machine translation model based on LSTM and made improvements by incorporating transformer-based methods. Specifically, we developed three English-French translation models using Opus-MT, T5, and bert2BERT[1]. We then compared the translation quality of these models. Finally, we extended the bert2BERT machine translation model to support multilingual translation by adding Hungarian.

## 1 Introduction

Machine translation (MT) has undergone rapid advancements with the development of neural network-based approaches. In this project, we enhanced a previously implemented large-scale language model (LSTM) for machine translation by transitioning to a Transformer-based architecture. Additionally, we expanded the training resources by collecting a new and significantly richer database compared to the original dataset.

The primary goal of this project is to build a robust multilingual machine translation model based on the Transformer architecture. To evaluate and refine our approach, we conducted a comparative analysis involving three models: the original LSLM-based model and two Transformer-based models leveraging T5 and bert2BERT architectures.

Our experiments were structured as follows:

1. We trained the three models on two distinct databases and performed comparative testing, resulting in six sets of outcomes. This analysis allowed us to examine the effects of the database quality and model architecture on machine translation performance.

2. We investigated the impact of training parameters by comparing full-parameter and adapter-parameter training strategies.

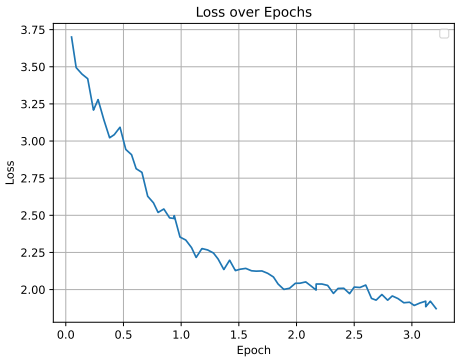
3. Finally, we extended the model's capabilities from bilingual Japanese-English translation to a multilingual framework, encompassing Japanese, English, French, Chinese, and German.

This project not only demonstrates the potential of the Transformer architecture in multilingual machine translation but also highlights the influence of training data, model selection, and parameter configurations on translation quality.

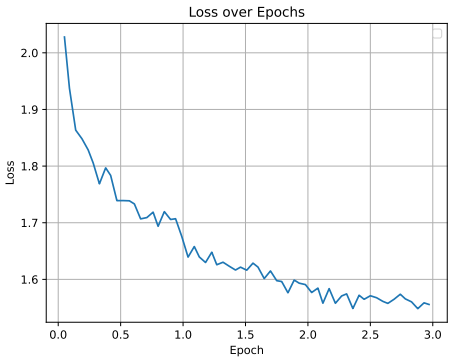
## 2 Method

### 2.1 T5

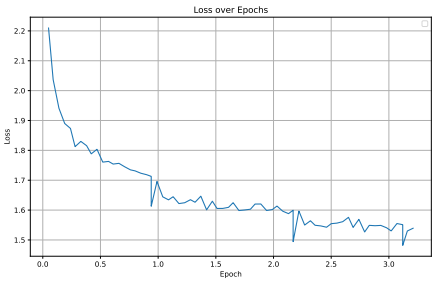
This project employs the T5 model for neural machine translation. The T5 framework reframes all NLP tasks into a text-to-text format, making it well-suited for sequence-to-sequence tasks like translation. The implementation leverages T5's encoder-decoder structure, pre-trained weights, and transfer learning capabilities to perform language translation between source language and target language.[4]



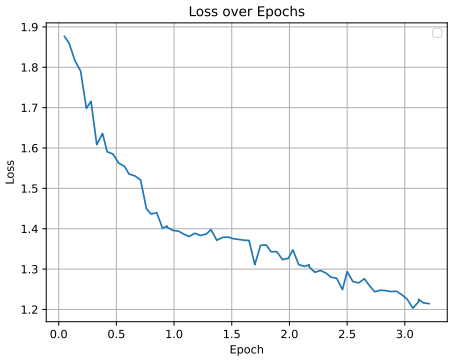

(a) LSTM




(b) T5




(c) Opus-mt




(d) bert2BERT

Figure 1: Train Loss

### Model Architecture

**1. Encoder:** The encoder processes the tokenized source text and generates contextual embeddings. Pre-trained weights from T5 were utilized, enabling transfer learning to leverage T5's general language understanding capabilities.

**2. Decoder:** The decoder generates the translated text sequentially by attending to both encoder outputs and its prior predictions through self-attention and cross-attention mechanisms. The decoder was fine-tuned to align the target language syntax and semantics with the source text.

**3. Task-Specific Input Format:** T5 uses a prefix format to specify the task, e.g., "translate English to French", which helps the model distinguish the task.

### Training

**1. Loss Function:** The model was optimized using a cross-entropy loss function computed over tokenized target sequences.

**2. Optimization:** The AdamW optimizer was employed, with learning rate scheduling using a cosine decay with warm-up steps.

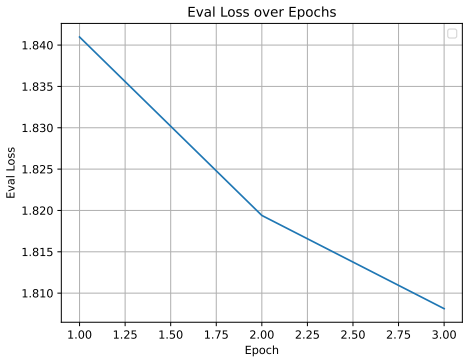
**3. Hyperparameters:** Learning rate: $2 * 10^{-5}$. Batch size: 16. Number of epochs: 16. Early stopping based on validation BLEU scores was used to halt training when overfitting was detected.

### Inference

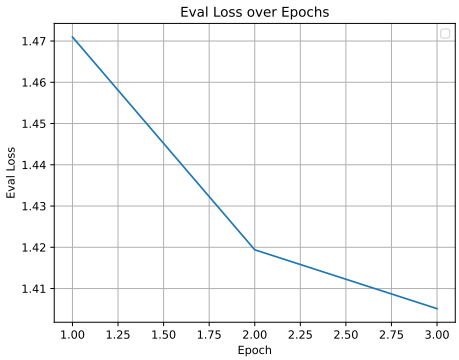
During inference, the T5 model generated translations using beam search decoding:

**Length Penalty:** A length normalization factor was applied to avoid biases toward shorter sentences.

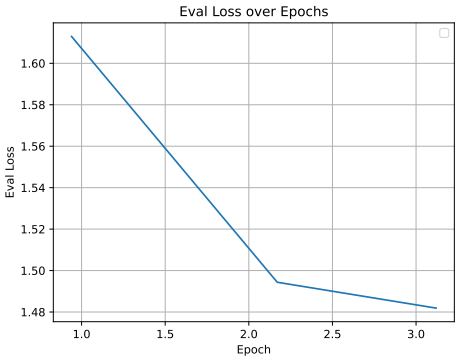
**Post-processing:** Tokenized outputs were detokenized and formatted to ensure readability.



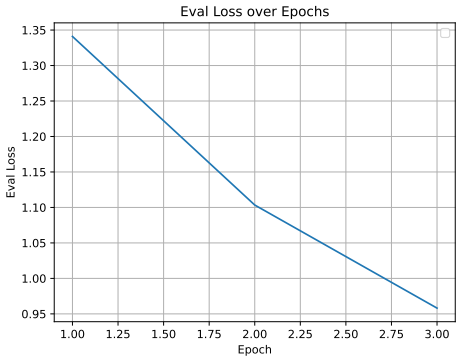

(a) LSTM




(b) T5




(c) Opus-mt




(d) bert2BERT

Figure 2: Evaluate Loss

### 2.2 bert2BERT

This project implements a neural machine translation system using the bert2BERT Transformer architecture. The objective is to translate texts from a source language to a target language. The methodology involves leveraging the self-attention mechanism and pre-trained bert2BERT embeddings for efficient sequence-to-sequence learning. The approach follows a standard encoder-decoder framework adapted for machine translation.[2]

#### Model Architecture

**1. Encoder:** The encoder uses a pre-trained BERT model to generate contextualized embeddings for the source sentences. Layer freezing: The initial layers of the pre-trained BERT model were frozen to retain the general language understanding while fine-tuning the upper layers for translation.

**2. Decoder:** The decoder is also a pretrained BERT. It employs masked self-attention to generate translations sequentially.

**3. Cross-Attention:** The decoder integrates encoder outputs through cross-attention layers, aligning the source sentence's contextualized embeddings with the target sequence during decoding.

#### Training

**1. Objective Function:** The model was trained using the cross-entropy loss function, with teacher forcing applied during the decoding process.

**2. Optimization:** The Adam optimizer was used with a learning rate scheduler employing a warm-up phase followed by inverse square-root decay.

**3. Regularization:** Dropout was applied to prevent overfitting. Label smoothing was introduced to mitigate overconfidence in predictions.

**4. Training Settings:** Batch size: 16 Number of epochs: 16 Early stopping criteria were used based on the BLEU score on the validation set.

#### Inference

During inference, beam search decoding was implemented to enhance the quality of the generated translations.

### 2.3 BART/OPUS-MT

This project implements a neural machine translation system using the OPUS-MT Transformer architecture, which is specifically designed for multilingual and domain-specific translation tasks. The model leverages the self-attention mechanism inherent in the Transformer framework to process language pairs efficiently. The approach adopts a standard encoder-decoder architecture, optimized for translation between English and French.[7]

#### Model Architecture

**1. Encoder:** The encoder in OPUS-MT is based on a Transformer architecture tailored for translation. It processes the source language sentences into contextualized embeddings using self-attention mechanisms. The encoder is pre-trained on large multilingual corpora, ensuring robust language representation across diverse domains.

**2. Decoder:** The decoder is a Transformer stack that generates target sentences token by token. It employs masked self-attention to handle the autoregressive nature of translation, predicting one word at a time while considering previously generated words.

**3. Cross-Attention:** The decoder incorporates cross-attention layers to integrate encoder outputs. This allows the decoder to focus on relevant parts of the source sentence embeddings when generating each word of the target sequence.

#### Training

**1. Objective Function:** The model was trained using the cross-entropy loss function, with teacher forcing applied during decoding to ensure stability and faster convergence.

**2. Optimization:** Training employed the Adam optimizer with a learning rate scheduler that combines a warm-up phase and inverse square-root decay to stabilize learning.

**3. Regularization:** Dropout was applied within the self-attention and feedforward layers to prevent overfitting. Label smoothing was introduced to reduce overconfidence in predictions, improving generalization.

**4. Training Settings:** Batch size: 16. Number of epochs: 16. Early stopping: Training was halted based on the BLEU score improvement on the validation set.

#### Inference

During inference, beam search decoding was utilized to generate high-quality translations by exploring multiple candidate sequences at each decoding step. This method balances exploration and exploitation, enhancing fluency and accuracy in generated translations.

## 3 Experiment

### 3.1 Evaluation

**BLEU (Bilingual Evaluation Understudy):** Evaluates the accuracy of translations by measuring the n-gram overlap between the model's output and reference translations. It is one of the most widely used quality assessment metrics in machine translation.

**TER (Translation Edit Rate):** Measures the error rate by calculating the edit distance between the generated translation and the reference translation. A lower TER indicates higher translation quality.

**METEOR:** A metric that incorporates semantic and syntactic information, particularly well-suited for evaluating translation tasks involving low-resource languages with rich linguistic structures.

The performance of the system was evaluated using standard NMT metrics, including BLEU, TER, and meteor.

### 3.2 Data

In this project, I chose OPUS Corpus [6] as the data source. OPUS Corpus is a public, massively parallel corpus covering a variety of language pairs and fields, and is suitable for research on machine translation tasks. 3

To ensure the high quality and consistency of the experimental data, I performed the following preprocessing steps on the data: **1. Cleaning:** Noisy data, such as non-English or French sentences, and repeated sentence pairs are removed to ensure the accuracy of the corpus. **2. participle:** Byte Pair Encoding (BPE) technology is used to segment the source language (English) and target language (French) sentences into words or subwords, thereby reducing the size of the vocabulary and improving the model's ability to process low-frequency words. **3. Alignment:** Ensure strict one-to-one correspondence between each sentence pair in the corpus to ensure the reliability of training and evaluation.

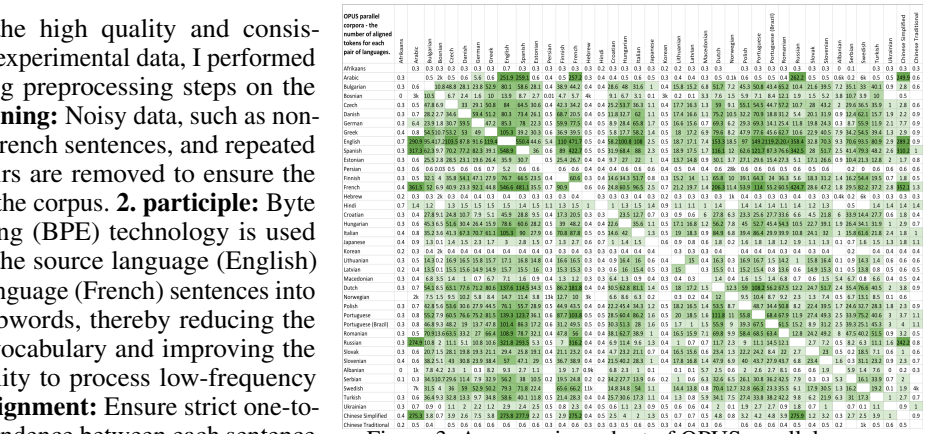


Figure 3: A comparison chart of OPUS parallel corpora

### 3.3 Result

#### 3.3.1 Train & Evaluation Loss

The training loss was recorded at the end of each epoch. As shown in Figure 1, the training loss steadily decreased as the number of epochs increased, indicating that the model was effectively learning from the training data.

To evaluate the model's generalization ability, we computed the evaluation loss on the validation set after each epoch. The evaluation loss is shown in Figure 2. Initially, the evaluation loss was higher, but it steadily decreased as the model trained, suggesting that the model was improving its performance on unseen data.

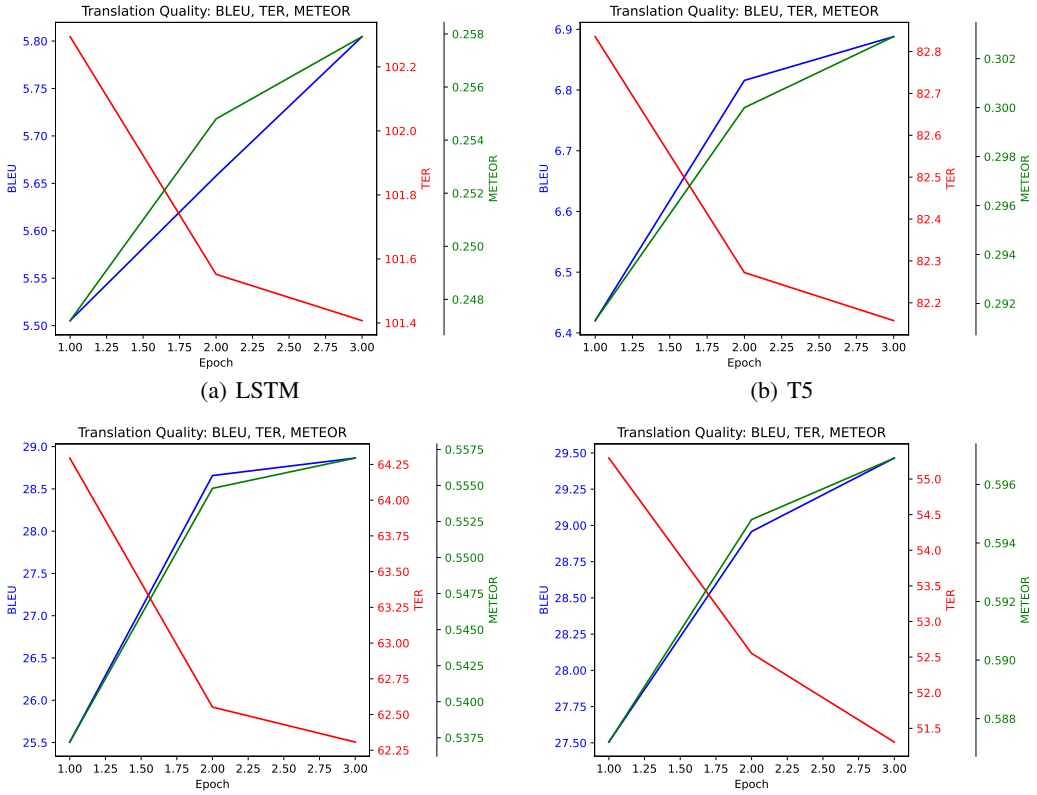
Based on the **Train_Loss** and **Eval_Loss**, the transformer-based machine translation models exhibit better properties compared to the LSTM-based models.

#### 3.3.2 Translation Quality

We use BLEU, TER, METEOR to evaluate the translation quality, and the results are shown in Figure 4. We observe the changes in translation quality for four models across training epochs, evaluated using BLEU, TER, and METEOR metrics.

For LSTM model, BLEU and METEOR scores gradually increase, indicating consistent improvement in translation quality, and TER scores gradually decrease, suggesting fewer translation errors over time. However, the improvements are limited, reflecting the slower convergence and constrained performance gains of LSTM.

For T5 model, BLEU and METEOR scores rise quickly in the early stages and achieve the highest values, showcasing superior translation performance, and TER shows a sharp decline, indicating T5 effectively reduces translation errors. Overall, T5 demonstrates strong convergence and exceptional translation capabilities.




(a) LSTM (b) T5

(c) Opus-mt (d) bert2BERT

Figure 4: Translation Quality

For Opus-mt model and bert2BERT model, BLEU and METEOR scores show a similar upward trend to T5, though slightly lower in absolute values. TER scores of Opus-mt decrease steadily, reflecting consistent error reduction, while TER scores of bert2BERT show a sharp downward trend, indicating a rapid reduction in translation errors. While their performance does not reach T5's level, they still exhibit strong translation quality.

These means T5 shows the best overall performance, achieving high BLEU and METEOR scores with a significant drop in TER, while LSTM performs the worst, with minimal improvements in translation quality. Opus-mt and bert2BERT deliver similar results, ranking below T5 but significantly outperforming LSTM. The results show that the transformer-based machine translation models yield higher translation quality, with T5 performing the best.

### 3.3.3 Example

We chose the sentence

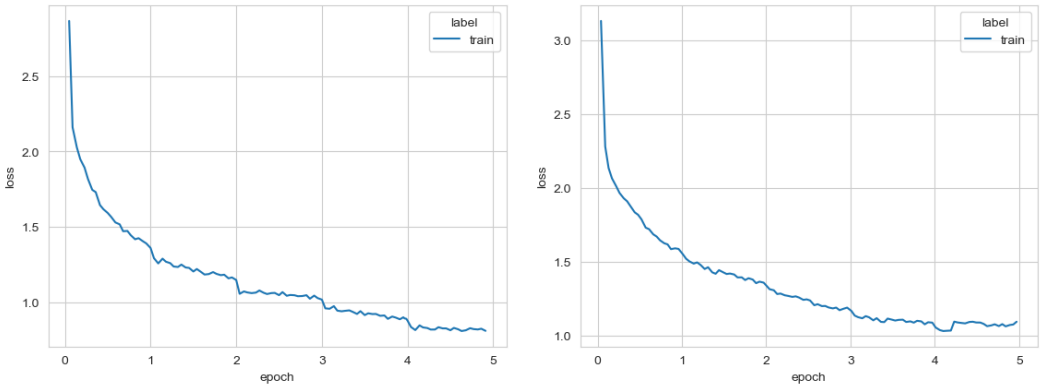
> ***Family, who gets it? Trained an AI model, and the accuracy ended up worse than random guessing!***

for translation and compared the translation results of different models. The result is showed below.

**LSTM:**

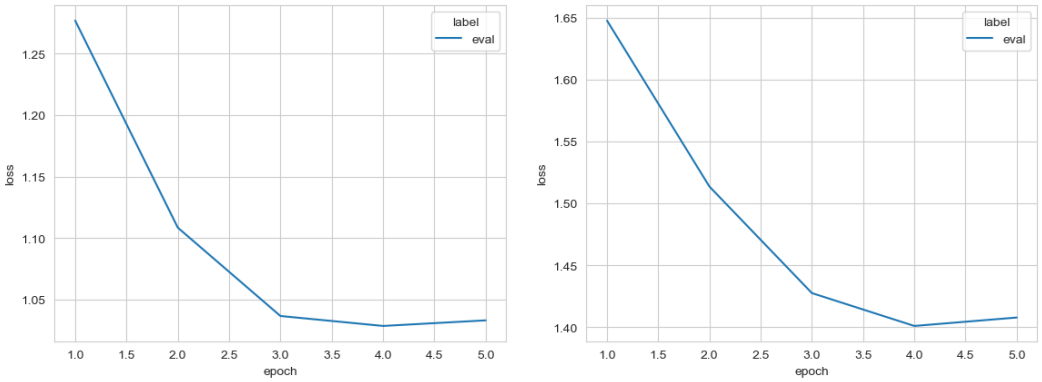
> ***La famille, qui l'a? J'ai formé un modèle d'IA, et la précision s'est avérée pire que des devinettes aléatoires!***

**T5:**




(a) Randomly Shuffled (b) En First and Hu Second

Figure 5: Multi-language Translation Train Loss




(a) Randomly Shuffled (b) En First and Hu Second

Figure 6: Multi-language Translation Evaluation Loss

***La famille, qui en a ? il a reçu un modèle d'IA, et l'exactitude finissait par pire que de deviner au hasard !***

**Opus-mt:**

***La famille, qui l'obtiendra? a formé un modèle d'IA, et l'exactitude a fini par être pire qu'une supposition au hasard!***

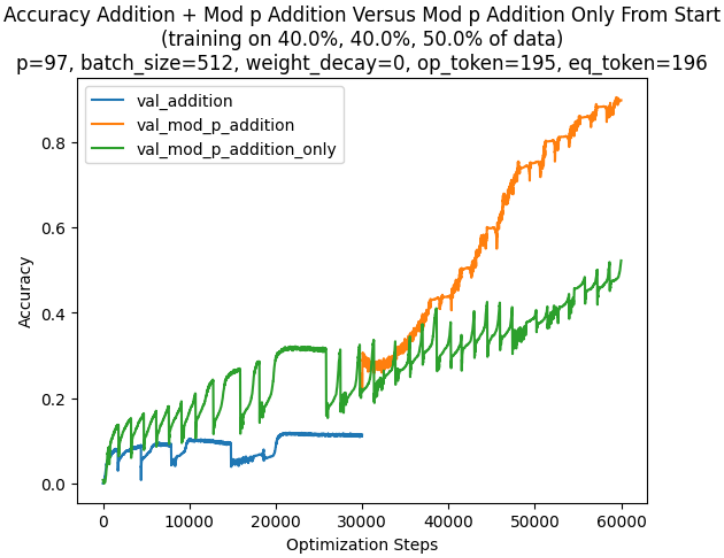
**bert2BERT:**

***Famille, qui comprend ? J'ai formé un modèle d'IA, et la précision s'est avérée pire qu'une supposition aléatoire!***

In the translation by LSTM, the subject and logic are relatively ambiguous. In the translation by T5, "*Il a reçu*" lacks logical consistency, seemingly introducing a new subject unrelated to the context. In the Opus-mt translation, "*A formé*" lacks a clear subject, making the sentence appear incomplete. In the bert2BERT translation, "*qui l'obtiendra*" reflects the interrogative tone of "who gets it" in the original sentence, with relatively coherent logic.

## 4 Mult-Language

In this part, we built a multilingual translation system based on the bert2BERT model, covering English to French and English to Hungarian translations.




(a)

Figure 7: Multi-step Learning

### 4.1 Motivation

We have 2 motivation for this reasearch:

1. We are interested in how MoE[5] or MoA[3] works, in another word, is MoE neccessary for multi-task if the tasks are similar to some extent, for example, multi-language translation.

2. The experiment shown in fig 7 is about how the order of data impacts the performance of naive transformer on addition and $mod_p$ addition tasks. From the result, we get the insight that language models might learn knowledge step by step, and the routine might be similar to human, which seems amazing. Motivated by the reasearch of multi-step learning, we are wondering how LLM makes it to learn how to translate English into other languages. Can human understand it like what we have presented below?

### 4.2 Experiment

#### 4.2.1 Data

In this section, we have 2 datasets: en-fr and en-hu[6].

We tried 2 methods to organize the datasets: **1. shuffle them randomly**. **2.En First and Hu Second**: By coressponding to learing them in the same time and in turn.

For each dataset, we have fine-tuned and evaluated them in the following sections.

#### 4.2.2 Evaluation

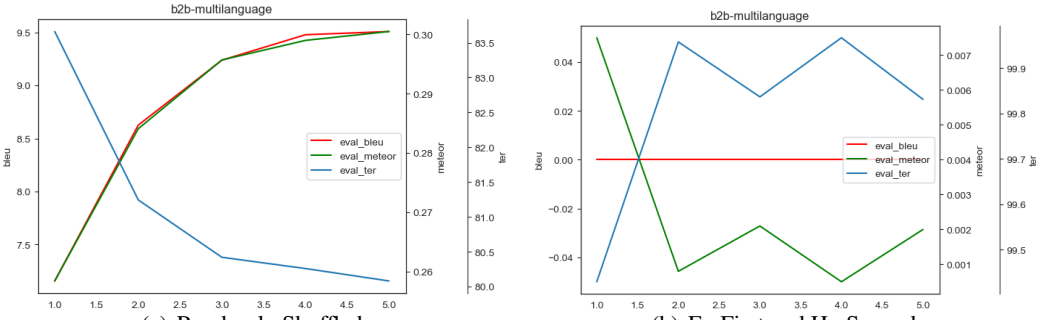
Like last section, we evaluate the our model in the same way,including BLEU, TER, METEOR. From the figure, both of them are learning something.

#### 4.2.3 Train & Evaluation Loss

Training loss, evaluation loss are presented in fig 5 and fig 6. Compare them with those in last part, we find that faced with multi-language task, bert2BERT does not perform as good as it performs on single-language and model trained by randomly shuffled data significantly outperforms the model trained by non-randomly shuffled data.

#### 4.2.4 Translation Quality

The result is showed in fig 8. Randomly shuffled one has the correct trend but non-randomly shuffled one has no correct trend at all.




(a) Randomly Shuffled (b) En First and Hu Second

Figure 8: Multi-language Translation Quality

#### 4.2.5 Example

**English to France:**

> ***Acceptez - vous, famille, qui se porte? Acceptez - vous un modèle à l ' art de l ' action,***

**English to Hungarian:**

> ***A család, ki tudja, hogyan jut eszébe az ember, és a számítás rosszabb, mint a kísérteties***

The bert2BERT fine-tuned by randomly shuffled figured out the basic Grammatical structure of both France and Hungarian, but it failed to figure out key words like family, AI. However, bert2BERT fine-tuned by Eng-Fra and Eng-Hun pairs in turn crushed during the training, which indicates that the model forget knowledge learned during pre-training, which is different from the naive transformer.

## 5 Conclusion

The transformer-based machine translation models outperform traditional LSTM architectures. Additionally, we have demonstrated that our model can be extended to a multilingual machine translation framework by randomly shuffled data.

## References

[1] Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. bert2bert: Towards reusable pretrained language models, 2021. URL `https://arxiv.org/abs/2110.07143`.

[2] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. Mixture-of-loras: An efficient multitask tuning for large language models, 2024. URL `https://arxiv.org/abs/2403.03432`.

[4] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

[5] Guijin Son, Sangwon Baek, Sangdae Nam, Ilgyun Jeong, and Seungone Kim. Multi-task inference: Can large language models follow multiple instructions at once? 2024. URL `https://arxiv.org/abs/2402.11597`.

[6] Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion

Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

[7] Jörg Tiedemann and Santhosh Thottingal. Opus-mt–building open translation services for the world. In *Proceedings of the 22nd annual conference of the European Association for Machine Translation*, pages 479–480, 2020.