# From Data Scarcity to Translation Fluency: A Multi-Staged Approach to Vietnamese-Lao and Lao-Vietnamese Machine Translation

Huy Ha [1*], Bach Le[1*], Quan Nguyen[1*], Tan Nguyen[1*], Ngoan Pham[2], Dac Nguyen[3], Dang Huynh[1†]

[1]Fulbright University Vietnam
[2]VNG Corporation
[3]Athena Studio

## Abstract

In this paper, we present two systems developed for participation in the translation shared task of the 10th International Workshop on Vietnamese Language and Speech Processing 2023 (VLSP23). These systems are designed for the two translation directions: Vietnamese to Lao and Lao to Vietnamese. Both systems are based on the Encoder-Decoder Transformer model for translation tasks and are implemented using OpenNMT_py without pre-trained weights. To enhance the performance of the systems and leverage the vast monolingual dataset, we attempted to employ backtranslation. However, this approach did not yield significant improvements for either system. We also performed model averaging, which resulted in a slight improvement in BLEU score, a widely used metric for evaluating translation quality. Our final submission systems attained BLEU scores of 19.3 and 19.1 for the Vietnamese to Lao and Lao to Vietnamese directions, respectively. The codebase for the systems can be found at https://github.com/FUVHasagi/VLSP23_FUV_gr2

## 1 Introduction

The translation shared task for VLSP23 focuses on the language pair Vietnamese and Lao, encompassing two separate tasks for both translation directions: Vietnamese to Lao and Lao to Vietnamese. This year's task presents a unique challenge due to the scarcity of training resources for these two languages and the restriction on using pre-trained weights. The limited availability of bilingual data classifies Vietnamese and Lao as low-resource languages. Moreover, advancements in low-resource machine translation are still nascent, and Lao is often overlooked in available datasets and models.

Our report delves into our approach to tackling both tasks, primarily focusing on the details of the provided dataset and the translation models we experimented with. We examine the bilingual data, conduct exploratory data analysis (EDA), and perform basic data cleaning/preprocessing. We replicate this process for monolingual data. Once the data is cleaned, we evaluate and experiment with various models to establish a baseline. Subsequently, we employ techniques like back-translation to leverage monolingual data to enhance model performance. This is followed by model averaging, also known as model ensembling. We also briefly overview our dockerization process for deploying the systems.

Our contribution is twofold. First, we examined and preprocessed bilingual and monolingual data to build a clean dataset which is shared for public use. Second, we established a multi-staged approach implementing back-translation and ensembling techniques to improve model representations and combine predictions for more accurate results.

The structure of this paper is as follows. Section 2 delves into a comprehensive overview of relevant literature. Subsequently, Section 3 examines the data utilized in our study, shedding light on its characteristics, composition, and the challenges associated with working with low-resource languages. Section 4 unveils the details of our methodology and outlines the training and validation processes. In Section 5, the outcomes are presented. Finally, Section 7 draws the paper to a close by encapsulating the key findings gleaned from our research.

## 2 Related works

Neural Machine Translation (NMT) witnessed a transformative phase with the emergence of early models, prominently characterized by the sequence-to-sequence (seq2seq) architecture. Differentiate themselves from the traditional methods, which are rule-based or statistical without the capability of learning directly from data for a more flexible and data-driven approach, these early implementations

* Equal contribution, alphabetical order.
† Corresponding author: dang.huynh@fulbright.edu.vn

with RNNs for facilitating the handling of sequential input and output setting up the basics for models are characterized with the architecture based on Encoder-Decoder models as the encoder processes input sequences, transforming them into fixed-size context vectors. At the same time, the decoder generates corresponding output sequences based on these context vectors, thus effectively applicable for variable-length input and output sequences (Sutskever et al., 2014). While these models laid the groundwork for neural approaches to machine translation, they also highlighted challenges such as information bottleneck impacting translation coherence for longer sentences and high computational and time consumption due to sequential processing. This prompted further advancements in architecture and training methodologies.

The introduction of attention mechanisms and transformer architectures addressed some of the limitations observed in these early models by allowing the model to selectively focus on different parts of the input sequence during decoding, addressing the limitation of capturing long-range dependencies. Moreover, transformer architecture further reduces the computational and time cost by enabling parallelization with positional encoding.(Vaswani et al., 2023) This golden age of transformers was marked by the appearance of translation models with superior performance compared to the previous ones, such as T5 (Raffel et al., 2023), BART (Lewis et al., 2019). By further development through leveraging a shared representation across language with additional tokens for language, the multilingual translation models such as mT5 (Xue et al., 2021), mBART (Liu et al., 2020), M2M100 (Fan et al., 2020) allow transfer learning, improve generalization to low-resource languages and contribute to the development of versatile models capable of handling diverse linguistic contexts.

Through the widespread use of platforms for pretraining, the training of new downstream translation tasks' performance benefits from good initialization, which facilitates transfer learning across tasks and further enhances the model's ability to capture rich contextual information. However, traditional transformer's architecture and approaches face multifaceted challenges that impact their effectiveness and robustness across diverse linguistic scenarios; for example, the incompleteness of the current evaluation system may not fully capture the nuances of linguistic diversity and context, leading to potential disparities between automated assessments and human judgments; the difficulty in domain adaptation and robustness in which the model get problems in the transition between domains or specialized vocabulary in which the model has not been explicitly trained, and nevertheless, the availability of high-quality data due to low resources languages lack sufficient parallel corpora causing lack of multilingual generalization.

Addressing these challenges has been a focal point in recent research. Techniques for mitigating data scarcity include leveraging transfer learning from resource languages. In contrast, pre-trained models can be fine-tuned on limited parallel corpora for low-resource languages or data augmentation with self-training, back-translation, or iteratively application of both to take advantage of the rich supply of monolingual data from both source and target language. Domain adaptation strategies have also emerged as crucial solutions to enhance the adaptability of NMT systems to specific contexts. Methods such as unsupervised domain adaptation, where models are trained on out-of-domain data and adapted to the target domain using domain adversarial training, have demonstrated effectiveness in improving translation quality in varied scenarios. Furthermore, developing appropriate evaluation metrics and benchmarks is vital to accurately assess the performance of NMT models, especially in low-resource settings. Recent studies emphasize the need for context-aware metrics that go beyond the traditional BLEU scores and consider factors including fluency, adequacy, and sensitivity to linguistic nuances in diverse languages. However, domain adaptation and evaluation systems methods are outside the scope of work in this situation.

In the context of the research challenge and the broader real-world scenario, Lao emerges as a language with inherent low-resource characteristics. Specifically, similar challenges and limitations are encountered within the Viet-Lao and Lao-Viet translation task domain. The need for more corpora in volume and quality across various subjects further compounds the difficulties associated with this linguistic task. Moreover, the specific exploration of this translation task in isolation from broader multilingual contexts has been limited, impeding the generation of high-quality datasets essential for future developmental endeavors. In light of these observations and recognizing the imperative

articulated within the competition framework, a critical need arises for acquiring a robust corpus and advancing more efficient and effective Neural Machine Translation (NMT) solutions dedicated to the Viet-Lao translation tasks.

## 3 Data discussion

### 3.1 Bilingual data

The bilingual dataset contains 100k sentence pairs. As we observed, the first 40k sentences have the topic of religion and news websites, and the next 60k sentences are 'common data' crawled web. In the first 40k pairs, exactly 10k pairs are duplicated. So we removed those, and there are 30k pairs left. Those 30k pairs are quite clean, so we move on with the last 60k pairs.

For the rest 60k pairs, we see that the data contains noises and even wrong translations (checked with the help of Google Translate). Those data samples might be crawled online, mostly from e-commerce and online advertisement websites. The word 'raw' means that sentences contain other unrelated languages, font errors, and even HTML format.

Hence, we decided to perform EDA on the bilingual dataset and then preprocess the data before feeding them into the model. We do not want too much noise in the data that might cause translation models to malfunction.

### 3.2 Monolingual data

Aside from the bilingual dataset, we are also equipped with two monolingual datasets. The Lao monolingual dataset comprises approximately 3 million sentences, while the Vietnamese monolingual dataset contains approximately 10 million sentences. This monolingual data, when harnessed effectively, can significantly enhance the performance of the translation system.

Later in this report, we will elaborate on our experiments with the monolingual datasets using backtranslation. To ensure data quality, we manually examined the Vietnamese monolingual data and observed that the topic aligns with daily news, similar to the bilingual data. This alleviates concerns regarding domain mismatch, allowing us to focus on data cleaning and adjustment. We also conducted a separate exploratory data analysis (EDA) session to trim the data to align with our training resources. For the Lao monolingual data, lacking expertise in this language, we performed a basic EDA session to trim the data similarly to the Vietnamese monolingual data.

### 3.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is essential in natural language processing (NLP) tasks, particularly when dealing with parallel datasets like the Vietnamese-Lao parallel dataset. EDA involves examining the data to understand its characteristics and identifying potential issues.

A thorough EDA of the Vietnamese-Lao parallel dataset revealed several issues, including duplicate sentence pairs, inconsistent spacings, font errors, emojis, and quoting discrepancies. Additionally, some sentence pairs exhibited extreme length disparities. We have eliminated redundant sentence pairs, standardized spacings, rectified font errors, expunged emojis, adjusted quoting conventions, and filtered out sentence pairs with extreme length ratios. Table 1 demonstrates some issues found in the original dataset.

By employing EDA and rigorous data cleaning procedures, the quality and consistency of the Vietnamese-Lao parallel dataset were significantly improved, laying the groundwork for more accurate and reliable NLP applications.

## 4 Methodology

### 4.1 Data preprocessing

As an initial step, we removed all duplicate pairs from the dataset. In instances where duplicates occurred within a single language (two identical Vietnamese sentences with two different corresponding Lao translations), we opted to retain these pairs as a single sentence, recognizing that one sentence can be translated into another language in multiple ways. Subsequently, we addressed special spacings, font errors, and emojis using for-loops and predefined dictionary rules, akin to iterating over all possible characters and employing template matching. However, for special spacings, we simply eliminated them from the sentences, while for font errors, we removed sentences containing font errors.

On the other hand, the removal of emojis is performed with the help of 'emoji' package (Wurster), we utilize the 'EMOJI_DATA' dictionary of the package. Furthermore, the rule for removing emojis is as follow: if a pair contains same emojis, then simply remove the emojis, else if only one sentence contains emojis, remove that pair. The next step is

| No. | Problems | Example (Vi) | Example (Lo) |
|---|---|---|---|
| 1 | Emojis | 👍 Thumbs up & Theo dõi + ... | 👍ໂປ້ & SUBSCRIBE + ... |
| 2 | Special spacing (NBSP, ZWSP) | Mô tả các tính năng kiến [ZWSP] trúc của SharePoint Server 2013. | ຄຸນສົມບັດໃຫມ່ໃນ SharePoint Server 2013 |
| 3 | Font errors | DIỄN ��ÀN | ພາສາລາວ |
| 4 | Irrelevant languages | Trang chủ / mở rộng lông mi / Misslamode long individual eyelashes Eyelash Extensions D Curl | ຫນ້າທໍາອິດ / ການຂະຫຍາຍຂົນຕາ / Misslamode Classic Eyelash Extensions 0.20mm C Curl |
| 5 | Quoting format | Một vấn đề đánh lạc hướng khác chính là " locavore " vừa được Từ điển New Oxford American Dictionary tôn vinh là " Từ của Năm " . | ບັນຫາລົບກວນອີກອັນ ໜຶ່ງ ແມ່ນ `` ທ້ອງຖິ່ນ & laquo; ໄດ້ ຮັບການຍົກຍ້ອງຈາກປຶ້ມ ວັດຈະນານຸກົມ New Oxford American ເປັນ & quot; ພຣະ ຄໍາ ຂອງປີ " . |
| 6 | Punctuations | 1. ID# ___(nhập của bạn ID ComPro # đây)___ 2. 123456Tiếp theo> >> Trang 1/166 | 1. ID # ___(ກະລຸນາໃສ່ຂອງ ທ່ານ ComPro ID # ທີ່ນີ້)___ 2. 123456ຕໍ່ໄປ> >> Page 1/166 |
| 7 | HTML entities and tags | Mã này:  Chữ nghiêng </ em> | ລະຫັດນີ້:  ເນັ້ງ  |
| 8 | Irrelevant numbers and phone numbers | Hệ nóng mở rộng (10 -6 / ℃) ‖6⊥12 ‖6⊥11 | ຄ່າສໍາປະສິດຮ້ອນຂອງການ ຂະຫຍາຍຕົວ (10 -6 / ℃) ‖6⊥12 ‖6⊥11 |
| 9 | URL | Trang web được tài trợ bởi iPadCasinoApp.com | ເວັບໄຊສະຫນັບສະຫນຸນໂດຍ iPadCasinoApp.com |
| 10 | Content repetition | 1. cờ bạc csgo 2. csgo cờ bạc trong tất cả | 1. ລະຫັດເງິນ CSGORacing 2. csgo, ຂໍ້ສະເຫນີ< |

Table 1: List of issues found in the origial dataset via exploratory data analysis. These issues are handled in a data cleaning process consisting of removing duplicates, special spacings, font errors, emojis, and adjusting quoting. We also filter out sentence pairs with extreme length ratios.

adjust quoting by using regular expression (regex), we simply replace the two single quotes (') and two backtick symbols (‘) by double quotes ("). Lastly, we perform sentence length ratio thresholding to keep only pair with length ratio (character level), $0.5 \leq \text{len}_{\text{vi}}/\text{len}_{\text{lo}} \leq 1.5$ as these thresholds covers approximately 90% of the dataset.

### 4.2 Tokenizer

We implemented Yasmin Moslem's (Moslem, 2023) workflow to create a Sentenpiece Unigram tokenizer model which was trained on 80K pair sentences, of which 68K pairs were from the filtered training set and 12K pairs were from the filtered validation set. The vocab size was set to have a limit of 50K.

| Language | Vocab limit | True vocab size |
|---|---|---|
| Vietnamese | 50K | 33K |
| Lao | 50K | 50K |

Table 2: Vocab size for Unigram model. The tokenizer model is trained only on the parallel dataset and was not trained on the monolingual dataset with 13M sentences due to limitted training resources.

We utilized the tokenizer model on the training and validation sets, which segmented the words within each sentence into subwords, separated by underscores in accordance with its vocabulary. During the inference stage, we employed the same tokenizer model to perform desubwording of the translated sentences, effectively removing the underscores separating the subwords and reconstructing them into complete sentences.

Initially, we intended to train the tokenizer on the entire monolingual dataset, encompassing 3 million Lao sentences and 10 million Vietnamese sentences. However, due to resource constraints, we were unable to train the tokenizer on 13 million sentences and opted to train it solely on 80,000 training and validation sentences.

### 4.3 Modelling

Both the Viet-Lao (VILO) and Lao-Viet (LOVI) models were built based on the Encoder - Decoder Transformer model for translation with OpenNMT-py implementation (Klein et al., 2017). The Encoder - Decoder Transformer model had 80M parameters, which was suitable for the amount of data available for training.

We trained the baseline model and all other variants using the Tesla P100 GPU 16GB on Kaggle, which allowed us to train models for extended periods of time without disconnecting. The two models were optimized using the Adam optimizer (Kingma and Ba, 2017) and a learning rate of 2 with Noam decay scheduler (Vaswani et al., 2023). We saved the model checkpoints every 5,000 steps. Detailed hyperparameters can be found in Table 3.

Model performance was evaluated using the BLEU score. For the last session, since we used the test set for training, we had to find another source of test data to compute BLEU score. We opted to using the Flores200 dataset (Team et al., 2022) to evaluate the last model.

| **Hyperparameters** | **Values** |
|---|---|
| Batch type | Tokens |
| Batch size | 4096 |
| Optimizer | Adam |
| Learning rate | 2 |
| Warmup steps | 1000 |
| Decay method | Noam |
| Dropout | 0.1 |
| Attention dropout | 0.1 |
| Checkpoint save interval | 5000 |

Table 3: Model hyperparameters and training setup

### 4.4 Training

We trained the models for a total of three major sequential sessions using different training steps and data. Most of the other parameters remained similar across all training sessions. Except for the baseline model (i.e., session 1), the remaining two sessions continued training the baseline model on the new dataset. The summary for the 3 sessions can be found in Table 4.

| **Sess.** | **Training data** | **#Sents.** | **#Steps** |
|---|---|---|---|
| 1 | (T) | 68K | VILO 85K<br>LOVI 32K |
| 2 | (BB) + (DBM) | 196K | 30K |
| 3 | (T) + (V) + (Te) | 82K | 30K |

Table 4: Details of three training sessions. Each session is characterized by a collection of training datasets containing a number of sentence pairs, and the number of training steps. Some notations: (T) is Train set; (BB) is Backtranslated Bilingual; (DBM) is Dual-Backtranslated Monolingual; (V) is Validation set; and (Te) is Test set.

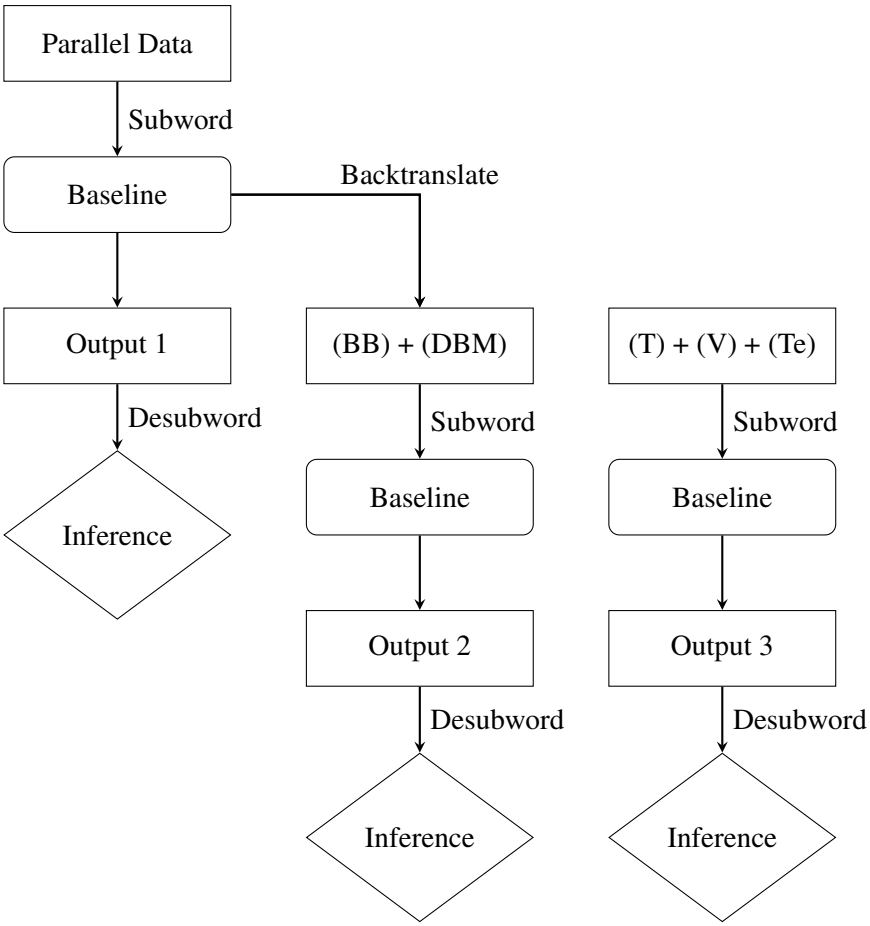


Figure 1: The entire workflow of creating and testing 3 models for each of the translation task. After using the trained baseline model to create backtranslated bilingual data (BB) and dual-backtranslated monolingual data (DBM), we feed the augmented data with the original parallel data into the baseline model to continue training. Lastly, the train set, validation set and test set are merged ((T) + (V) + (Te)) and fed to the baseline model for optimal performance (see Table 4).

For each task, the first session was to train the baseline model on the filtered training (T) dataset. We trained the VILO model for 85K steps and the LOVI model for only 32K steps. The imbalance between the two tasks was due to a lack of training resource and time.

The second session used a combination of the original training data, backtranslated bilingual (BB) data and dual-backtranslated monolingual (DBM) data for each language. This adds up to a total of 196K sentence pairs, of which 60K were translated monolingual sentences. Both models were trained for 30K steps.

The last session was to continue training the baseline model on the training set merged with the validation set (V) and test set (Te) for maximized accuracy. We also trained the two models for 30K steps each.

### 4.5 Backtranslation

We applied backtranslation twice in order to boost the performance of the system. The bilingual dataset was translated using our baseline models and the translated sentences were merged with the original training set. The source language was concatenated with the translated version, i.e., 68K original Lao sentences with 68K Lao sentences translated from Vietnamese using the VILO model, and the target language was kept the same, i.e., 68K Vietnamese sentences was only duplicated to create the parallel dataset for training the LOVI model. After using backtranslation on the bilingual dataset, we have the first augmented dataset with 136K pairs of sentences.

In addition, we also applied dual-level backtranslation. A sample of 30K sentences from the monolingual dataset was chosen and translated for two times using both the LOVI and VILO model.

For LOVI, the source data contained 136K augmented bilingual Lao sentences, 30K original Lao sentences and 30K Lao sentences that went through 2 translations (Lao ⇒ Viet ⇒ Lao) which summed up to 196K Lao sentences. The target data is duplicated using the 30K Lao sentences translated from Viet using the VILO baseline model.

### 4.6 Averaging Models

Using the averaged weights of different checkpoints is a strategy to improve model performance (Gao et al., 2022). Models checkpoints were averaged during inference using OpenNMT-py built-in average models script.

## 5 Results and discussion

For the Vietnamese - Lao translation task, the baseline model, which was the average from the 80K and 85K steps checkpoints, had the best performance with a BLEU score of 19.3. Implementing backtranslation and merging train, validation and test set for training yielded significantly worse results. The results can be found in Table 5.

Upon further inspection, we observed that the DBM data had not been desubworded correctly as there were still many underscores and <unk> tokens. We believe that this was due to the tokenizer being trained on such a small dataset which was not generalized enough.

As for the third model, we suspected that the Flores200 dataset had a completely different distribution compared to the given dataset. This might have led to the low BLEU scores.

The models in the Lao - Vietnamese translation tasks also observed similar behaviour to the Vietnamese - Lao models, with the baseline model having the best performance based on BLEU scores. However, the last LOVI model performed much better than that of the VILO task. The details of the models' performances can be found in Table 6

A thorough investigation into the vocabulary of both languages revealed that words in Lao were often divided into phrases of multiple characters instead of single characters. This could have led to a flawed tokenizer for Lao, leading to the models' poor performance.

| Model | Description | BLEU |
|---|---|---|
| 1 | Baseline | **19.3** |
| 2 | (BB) + (DBM) | 11.97 |
| 3 | (T) + (V) + (Te) | 0.29 |

Table 5: BLEU scores for VILO models.

| Model | Description | BLEU |
|---|---|---|
| 1 | Baseline | **19.1** |
| 2 | (BB) + (DBM) | 11.19 |
| 3 | (T) + (V) + (Te) | 5.35 |

Table 6: BLEU scores for LOVI models.

## 6 Deployment

This section is about how we deploy the docker images for submission. Because our submission uses an Open-NMT structure and this module uses bash script to run the training to test the model, our docker image is basically built from a bash script file.

Therfore, we approach the problem of transferring the input test file into the docker image by using the Python's library `os` to load and test the model by setting an initial input file and output file in the container. For the input file (`/app/input.file`), we were able to write that file with the test input file, and for the output file (`/app/UN.<target language>.translated.desubword`) we did the same process and later on, write that file with the translated text from our model.

As for the requirements of the docker images, we make use of Open-NMT.py version 3.4 and Python's sentencepiece library and (`https://github.com/ymoslem/MT-Preparation.git`) for input text pre-processing and output text post-processing.

## 7 Conclusion and future work

To sum up, for the Machine Translation task in VLSP23, we utilized OpenNMT-py's implementation for the Encoder - Decoder Transformer model as well as Sentencepiece's Unigram tokenizer to train three translation model versions for Vietamese - Lao and Lao - Vietnamese. We applied backtranslation and model weight averaging to optimize performance.

For future work, we plan to investigate the impact of different backtranslation strategies, experiment with alternative tokenizer models, fine-tune the Encoder-Decoder Transformer model, and integrate pre-trained language models, such as BERT and BART, into our machine translation framework could potentially improve the semantic understand-

ing of the source language and produce more fluent translations.

## References


Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Yingbo Gao, Christian Herold, Zijian Yang, and Hermann Ney. 2022. Revisiting checkpoint averaging for neural machine translation.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Yasmin Moslem. 2023. MT-Preparation. Original-date: 2020-11-09T06:13:25Z.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Taehoon Kim Wurster, Kevin. emoji: Emoji for Python.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer.