Training and Benchmarking Neural Machine Translation Models

Ethan Mathieu & Shankara Abbineni
Yale University
New Haven, CT
{ethan.mathieu, shankara.abbineni}@yale.edu

Abstract

In this project, we ask two questions: what are the gains to fine-tuning general langauge models on translation; and can general language models, when fine-tuned, perform better on translation tasks than a model trained solely for translation. As such, we train the DeLighT transformer model for English-to-French translation and compare its BLEU performance to other neural machine translation models which we fine-tune. We find that fine-tuned general language models can perform better than language-specific models. Additionally, we build a NextJS web application to allow end users to experiment with the different models and view their performance.[1].

1 Project Motivation

The task of machine translation refers to the use of computational algorithms and neural networks to translate between human languages. As the world becomes more globalized, reducing language barriers is integral for fostering collaboration and innovation. Therefore, the value in having a capable machine translation model is undeniable—especially if these models are able to achieve levels of accuracy that rival human translators. In order to construct more capable models in the future, we must first find the optimal model type. As such, in this project, we raise questions regarding the efficiency and accuracy of models. One such question is what the performance benefits are from fine-tuning general language models—which are exposed to various styles of language—on a specific translation task (i.e. one specific language to another specific language)? Another question that we ask is how do the performance characteristics of these fine-tuned models compare to a model trained from the ground-up on a translation dataset? Will the general language models that are fine-tuned perform better on a single translation task—as they may have a greater understanding of linguistic structure across multiple languages that may relate to the translation task at hand—or will the model that is trained solely for the specific, single translation task perform better due to it having maximal exposure to the two languages in question? In hopes of finding answers to these questions, we train a DeLighT model which focuses purely on English-to-French translation and compare its performance (using BLEU scores) to general models which have been fine-tuned on an English-to-French dataset. In doing so, we seek to explore some of the most promising methods of machine translation in this paper and evaluate their relative merits.

Finally, in order to extract the full value of these models in relation to translation tasks, we implement a web-based application through which a user may access and utilize the DeLighT model, as well as the other pre-trained and fine-tuned models, on English-French translation tasks. The application consists of a graphical user interface which allows the user to see the differences in translations between models on the same text. With this, our project not only further explores a research question, but also serves as a practical solution for reducing language barriers and identifying models most capable for translation tasks.

[1] Project code on GitHub

The code for this project is available on GitHub at https://github.com/emath12/machine-translation/tree/main.

2 Related Work

Various models and methods of neural machine translation have been proposed in prior literature. Many neural machine translation models prior to the publication of the transformer architecture from Vaswani et al. [2017] relied heavily on recurrent neural networks (RNNs) or long short-term memory (LSTM) architectures. For instance, the use of an encoder-decoder RNN was explored by Sutskever et al. [2014], achieving record performance on a sequence-to-sequence translation task. Similarly, the work of Cho et al. [2014] explored the use of LSTMs for machine translation and further expanded a model's capability of understanding linguistic relations separated by large amounts of text within an input sentence. Currently, many of the most capable neural machine translation utilize the transformer architecture presented by Vaswani et al. [2017]. One such architecture is the DeLighT model, as presented by Mehta et al. [2020]. The main benefit of the DeLighT model is that it replaces traditional transformer blocks with DeLighT blocks which perform a DeLighT transformation. A DeLighT transformation uses group linear transformations (GLTs) to learn local representations by deriving the output from a specific portion of the input, making the transformation more efficient than commonly used linear transformations. Then, to learn global representations, the DeLighT transformation utilizes feature shuffling to share information between different groups within the GLTs. The DeLighT transformations are then stacked and integrated into DeLighT blocks. The second benefit of the DeLighT architecture is that it employs block-wise scaling to construct a network with variably-sized DeLighT blocks. More specifically, to reduce parameter counts and contribute to the model's lightweight nature, the model allocates shallower and narrower DeLighT blocks near the input, and deeper and wider DeLighT blocks towards the output. Using the DeLighT model, Mehta et al. [2020] were able to present strong performance results on single language translation tasks, such as English-to-French and English-to-Spanish. In experiments, Mehta et al. [2020] demonstrate that DeLighT is able to match the performance of baseline transformers using 2 to 3 times fewer parameters on average.

Past attempts at fine-tuning pre-trained models on language translation tasks have shown that the fine-tuning process can often lead to increases in translation performance. Zhu et al. [2020] demonstrate this by using a BERT model for neural machine translation. The paper proposes a new algorithm titled the "BERT-fused model" which first uses BERT to extract representations for an input sequence, and then fuses these representations with each encoder and decoder layer of an NMT model [Zhu et al., 2020]. In their results, Zhu et al. [2020] found that the BERT-fused model was able to achieve BLEU scores roughly 7 percent higher than the standard transformer model. In a similar fashion, Yang et al. [2020] propose a concerted training framework, dubbed the"CTnmt" model, to better fine-tune existing models such as BERT and GPT2 on language translation tasks. The proposed CTnmt model of Yang et al. [2020] is comprised of three key methodologies: (1) asymptotic distillation, aimed at preserving prior pre-trained knowledge within the NMT model; (2) a dynamic switching gate, strategically designed to prevent the sudden loss of previously acquired knowledge; and (3) a learning rate adjustment strategy, tailored to adapt the learning pace based on a predetermined schedule. With this specific approach to fine-tuning, the CTnmt model is able to present gains to the BLEU score of roughly 5 percent [Yang et al., 2020].

3 Approach

Our approach to this project consists of three components: (1) training a DeLighT model purely for English-French translation; (2) obtaining general language models which have been exposed to multiple languages and fine-tuning them on English-French translation datasets; and (3) constructing a web interface for end users to interact with and compare the models. All datasets can be found on HuggingFace merely by searching the dataset name.

3.1 Training the DeLighT Model

For this project, we decided to use the DeLighT model as the benchmark for a model trained specifically to translate from English to French. The DeLighT model was chosen because of its unique

architecture design, which as mentioned in the prior section, allows for the model to perform strongly on translation tasks while using far less parameters than other transformer models [Mehta et al., 2020]. In order to train a model under the compute limitations we faced, the DeLighT model we use in this project contains only two encoder layers and two decoder layers—making our DeLighT model roughly a quarter of the size of the model trained by Mehta et al. [2020]. The training data used to train our DeLighT model is the WMT2014 English-French dataset, which countains roughly 40.8 million translations of English and French sentences and phrases. Each row of the dataset contains a dictionary with "en" and "fr" as keys to access the respective translation. For example, one such row in the dataset is `{"en": "Please rise, then, for this minute' s silence.", "fr": "Je vous invite à vous lever pour cette minute de silence."}`. The dataset is processed using the Fairseq library to develop the sequences used for training the model [Ott et al., 2019]. Following this, we trained the model on the WMT2014 English-French dataset using an NVIDIA A5000. We use Adam to minimize cross entropy loss with a label smoothing value of 0.1 during training. Furthermore, we use a learning rate of $1\text{E}-5$ and trained on 2000 steps. The learning rate also had 100 steps of linear warm-up followed by linear decay. Also, the batch size used was 32. While constructing and training this model, we closely followed the specifications outlined and made use of much of the code provided by Mehta et al. [2020]; however, changes were made to the depth and size of the model in order for it to be trainable given our compute constraints.[2] In order to compare our DeLighT model with the other models that we fine-tuned, we utilized the BLEU score as the primary evaluation metric.

3.2 Fine-tuning General Models

Fine-tuning pre-trained language models has become a popular approach for adapting them to specific downstream tasks. By leveraging the knowledge captured in these large models during pre-training, fine-tuning allows us to efficiently create specialized models which can perform better on specific tasks in translation. The particular base transformer models we chose to fine-tune were Google's T5, Facebook's base BART model, and Microsoft's Marian. The T5 makes use of a causal decoder and focuses on training the base model to text-to-text tasks (i.e. commands). Facebook's BART model is a sequence-to-sequence transformer model trained as a denoising autoencoder, meaning it can take a text sequence as input and produce a different text sequence as output. While the T5 and BART models are general language models, Marian is a standard transformer model that was trained primarily for translation. By including the Marian model in our experimentation, we can obtain a control allowing us to understand the how fine-tuning impacts models based on the nature of their initial training. Each of these base models were fine-tuned using a T4 GPU from Google Colab. The learning rate used was $2\text{E}-5$. The batch size for fine-tuning was 32, and the batch size for evaluation was 64, running for a total of three epochs. Some minor pre-processing was done to transform the dataset for consumption by a standard transformer library tokenizer.[3] The models were fine-tuned using the same training subset of the ISWLT2017 and KDE4 English-French datasets, ensuring that during evaluation, they are exposed to sentences and phrases which they have not seen before (further information on training is presented in the following subsection).

3.3 Evaluation Metrics and Evaluation Datasets

Due to its simplicity and effectiveness in measuring the similarity between machine-generated translations and human reference translations, the BLEU score is a common metric used amongst researchers in evauating machine translation models. BLEU computes a geometric mean of n-gram precision, penalizing overly short translations and favoring translations that align well with reference translations. Since it is a metric that is quick to obtain and relatively reliable in assessing translation quality, it was deemed a suitable choice for evaluating our models. BLEU is mathematically defined as

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \cdot \log p_n\right)$$

[2]Credit to the base code provided by the authors of DeLighT.

[3]Fine-tuning approach based on HuggingFace tutorial.

where BP is the brevity penality defined by

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases}$$

Note that p_n is the precision of n-grams (sub-strings of length n) between the candidate translation and reference translations, w_n is the weight assigned to the n-gram precision, c is the length of the candidate translation, and r is the length of the closest reference translation.The metrics for our models performances are presented in Section 4.

As mentioned earlier, the WMT2014 English-French dataset, containing a large number of parallel sentences, was used to train the DeLighT model, providing it with a broad foundation in English-French translation. In contrast, subsets of the ISWLT2017 English-French and KDE4 English-French datasets were employed for fine-tuning the Marian and T5 models. The ISWLT2017 English-French dataset consists of 220,400 rows of parallel sentences, while the KDE4 English-French dataset comprises 210,173 rows. Each dataset is structured as a dictionary, with "en" and "fr" as keys corresponding to the English and French sentences, respectively. This dictionary format allows for easy access and mapping between the source (English) and target (French) sentences during the fine-tuning process.

To ensure accurate evaluation of the models' performance on unseen data, the ISWLT2017 and KDE4 datasets were split into training and testing subsets. The training subsets, comprising of a fixed 90 percent of each dataset, were used to fine-tune the Marian and T5 models; on the other hand, the testing subsets (containing a fixed 10 percent of each dataset), which were not exposed to the models during fine-tuning, served as held-out data for evaluating the models' translation quality using BLEU scores. By using separate training and testing subsets, we can obtain reliable and unbiased BLEU statistics that accurately reflect the models' performance on unseen data. This approach ensures that the models are not merely memorizing the training data but are actually learning to generalize and produce high-quality translations for new, previously unseen sentences. All of the datasets were obtained through Hugging Face: the WMT2014 dataset, the IWSLT2017 dataset, and the KDE4 dataset can be found on Hugging Face and can easily be used through the Datasets library.

3.4 Constructing a Web Interface

A web interface was constructed using the latest version of NextJS. On the backend, FastAPI facilitates the calls to each model. All of our models trained or fine-tuned in this paper were uploaded to HuggingFace and given its own FastAPI endpoint, saturated with the HuggingFace API, for the frontend interface to call.

The frontend displays the BLEU results for the selected models and allows the user to type in their own English queries to the models.

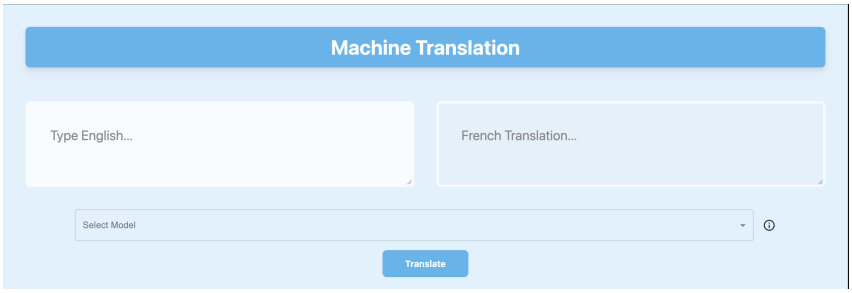

Figure 1: Image of the web interface.

4 Results

The performance results of the Marian (base and fine-tuned) models, T5 (base and fine-tuned) models, and DeLighT models are presented in Tables 1 and 2. Table 1 contains the BLEU scores for the models when evaluated on the ISWLT2017 English-French dataset. Similarly, Table 2 holds the BLEU scores for all of the models after evaluation on the KDE4 English-French dataset.

Model	BLEU Score
Marian	38.825
Marian (fine-tuned)	40.884
T5	1.177
T5 (fine-tuned)	35.586
BART	1.128
BART (fine-tuned)	31.878
DeLighT	32.114

Table 1: BLEU scores from evaluating models on the ISWLT2017 English-French dataset.

Model	BLEU Score
Marian	39.265
Marian (fine-tuned)	47.283
T5	5.002
T5 (fine-tuned)	43.718
BART	8.059
BART (fine-tuned)	41.204
DeLighT	35.270

Table 2: BLEU scores from evaluating models on the KDE4 English-French dataset.

4.1 Performance Benefits from Fine-tuning

The results presented in Tables 1 and 2 clearly demonstrate the performance benefits of fine-tuning both the Marian and T5 models on the ISWLT2017 and KDE4 English-French datasets. Fine-tuning allows the models to adapt to the specific domain and style of the target datasets, leading to significant improvements in translation quality as measured by the BLEU score. Looking at Table 1, which shows the results on the ISWLT2017 dataset, the base Marian model achieves a respectable BLEU score of 38.825; however, after fine-tuning, the Marian model's performance improves to 40.884, representing a notable increase of 2.059 BLEU points. This improvement indicates that the fine-tuned Marian model has learned to better handle the intricacies and nuances of the ISWLT2017 dataset, resulting in more accurate and fluent translations. In comparison, the base T5 model initially performs poorly on the ISWLT2017 dataset, with a BLEU score of only 1.177. This low score suggests that the base T5 model, without fine-tuning, heavily struggles to generate high-quality translations for this specific dataset. Upon fine-tuning, though, the T5 model's performance drastically improves, reaching a BLEU score of 35.586. This remarkable increase of 34.409 BLEU points highlights the importance of fine-tuning in adapting the T5 model to the ISWLT2017 dataset, allowing it to produce translations that are far more accurate and coherent. Examining the BART model, we once again see massive improvements to the BLEU score—akin to what was seen with the T5—after fine-tuning. The BART model has an initial BLEU score of 1.128 on the ISWLT2017 dataset; but after fine-tuning, the model's BLEU score increased by 30.75 BLEU points to 31.878. As such, the gains to fine-tuning showcased by BART are incredibly strong.

In Table 2, which presents the results on the KDE4 dataset, a similar trend can be observed. The base Marian model achieves a BLEU score of 39.265, while the fine-tuned Marian model reaches an impressive 47.283, an improvement of 8.018 BLEU points. This substantial increase in performance demonstrates the effectiveness of fine-tuning in enabling the Marian model to better capture the characteristics of the KDE4 dataset. Furthermore, the base T5 model performs extremely poorly on the KDE4 dataset, with a BLEU score of only 5.002. However, after fine-tuning, the T5 model's performance skyrockets to a BLEU score of 43.718, an astonishing increase of 38.716 BLEU points. This dramatic improvement underscores the crucial role of fine-tuning in allowing the T5 model to generate high-quality translations for the KDE4 dataset.

The results presented in Tables 1 and 2 provide strong evidence for the performance benefits of fine-tuning general language models on specific translation tasks. For the general language models, such as T5 and BART, the gains to fine-tuning are massive. Even the Marian model experienced a decent level of improvement, which is impressive considering it is a model that was trained by Microsoft with translation in mind. Thus, fine-tuning models that have already been trained with

translation in mind may see respectable performance uplifts from fine-tuning on specific styles of language that may not be present in their training data. These findings highlight the importance of fine-tuning in developing high-performance machine translation systems tailored to specific domains and styles.

4.2 Comparison of Language-specific Models to General Language Models

The results in Tables 1 and 2 allow for a comparison between the language-specific DeLighT model and general language models, such as BART, T5, and Marian, along with their fine-tuned variants.

On the ISWLT2017 dataset (Table 1), the DeLighT model achieves a BLEU score of 32.114, which is lower than the fine-tuned Marian model's score of 40.884 and the fine-tuned T5 model's score of 35.586. However, DeLighT outperforms the base Marian model (38.825), the base T5 model (1.177), and both the base (1.128) and fine-tuned (31.878) BART models. This suggests that while DeLighT may not reach the same level of performance as the best fine-tuned general language models, it still provides competitive results and outperforms some of the base and even fine-tuned models. On the KDE4 dataset (Table 2), the DeLighT model obtains a BLEU score of 35.270, which is lower than the scores of the fine-tuned Marian (47.283), T5 (43.718), and BART (41.204) models. That being said, though, DeLighT surpasses the performance of the base T5 (5.002) and BART (8.059) models, while being slightly behind the base Marian model (39.265).

These results indicate that language-specific models like DeLighT can provide competitive performance compared to general language models, especially when considering base models without fine-tuning; however, fine-tuned general language models tend to outperform language-specific models, highlighting the importance of fine-tuning for achieving state-of-the-art performance in machine translation tasks. It is worth noting that language-specific models like DeLighT have the advantage of being specifically designed and trained for a particular language pair, which can lead to more efficient and targeted training. On the other hand, general language models offer greater flexibility and can be fine-tuned for various language pairs and domains, making them more versatile. With respect to the original question of whether a fine-tuned general language model has better performance than a specific model trained for a translating between two specified languages from the ground-up, our experiments indicate that language-specific models like DeLighT provide competitive performance and have the advantage of targeted training, but that fine-tuned general language models consistently achieve higher BLEU scores across both datasets. This suggests that fine-tuning general models is a more effective approach for obtaining state-of-the-art performance in machine translation tasks compared to training language-specific models from scratch.

5 Discussion

In this project, we explored the performance of various neural machine translation models, including the DeLighT model which was trained specifically for English-to-French translation, and general language models like T5, BART, and Marian which were fine-tuned on English-French datasets. Through our experiments, we found that fine-tuning general language models led to significant improvements in BLEU scores compared to the base models. The fine-tuned T5 and Marian models achieved the highest BLEU scores on both the ISWLT2017 and KDE4 datasets. While the DeLighT model provided competitive performance, especially compared to the base general language models, it was ultimately outperformed by the fine-tuned models. These results suggest that fine-tuning large, pre-trained language models is an effective approach for obtaining state-of-the-art performance on machine translation tasks.

There exist some limitations to our approach that should be noted. First, due to compute constraints, the DeLighT model trained for this paper had to be smaller in size. It's possible that with more compute resources, a larger DeLighT model could achieve even better performance. Additionally, we only explored fine-tuning on two datasets (ISWLT2017 and KDE4). Future work could involve fine-tuning on a wider variety of datasets spanning different domains to further assess the generalizability of the fine-tuned models. Finally, while BLEU score is a commonly used metric for evaluating machine translation, it has some known limitations, such as poor correlation with human judgement. Future work could incorporate additional evaluation metrics like chrF (CHaRacter-level F-score) or human evaluations to get a more holistic assessment of translation quality.

Contribution Statement

There was even contribution between both authors for the completion of this project. Ethan worked primarily on the model fine-tuning and web interface aspects of this project. Shankara worked more on training the DeLighT model and aided in conducting the evaluations. These tasks were not strictly completed by each person alone—rather each of us had input and worked on what the other person was assigned with as well. Both authors contributed evenly to the research required to complete the project. Lastly, the written report was compiled by both authors with equal contribution from both.

References

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

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL `http://arxiv.org/abs/1409.3215`.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL `http://arxiv.org/abs/1406.1078`.

Sachin Mehta, Marjan Ghazvininejad, Srinivasan Iyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Delight: Very deep and light-weight transformer. *CoRR*, abs/2008.00623, 2020. URL `https://arxiv.org/abs/2008.00623`.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*, 2020.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. Towards making the most of bert in neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9378–9385, 2020.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Reproducibility checklist - Ethan and Shankara

* Please make sure these points are addressed in your report submission
* Please copy this and replace the ☐ with a ✔ for the items that are addressed in your report/code submission
* Please complete this report, attach it to your final project report as the last page and then submit.

Model Description, algorithm, Mathematical Setting:

- ✔ Include a thorough explanation of the model/approach or the mathematical framework

Source Code Accessibility:

- ✔ Provide a link to the source code on github.
- ✔ Ensure the code is well-documented
- ✔ Ensure that the github repo has instructions for setting up the experimental environment.
- ✔ Clearly list all dependencies and external libraries used, along with their versions.

Computing Infrastructure:

- ✔ Detail the computing environment, including hardware (GPUs, CPUs) and software (operating system, machine learning frameworks) specifications used for your results.
  - (Example statement 1: the model was fine-tuned using a single T4 GPU on colab.
    Example statement 2: we ran inference of Llama 70B using 4 Nvidia A5000 GPUs)
- ✔ Mention any specific configurations or optimizations used.
  - (Example: We used a quantized version of Llama with int8.
    Example 2: We used the regular float32 representation.)

Dataset Description:

- ✔ Clearly describe the datasets used, including sources, preprocessing steps, and any modifications.
- ✔ If possible, provide links to the datasets or instructions on how to obtain them.

Hyperparameters and Tuning Process:

- ✔ Detail the hyperparameters used and the process for selecting them.
  - (Example: The model was fine-tuned using a batch size of 16, learning rate of 1e-5, and trained on 1000 steps with 100 steps of learning rate linear warm up with linear decay)

Evaluation Metrics and Statistical Methods:

✔ Clearly define the evaluation metrics and statistical methods used in assessing the model.

Experimental Results:

✔ Present a comprehensive set of results, including performance on test sets and/or any relevant validation sets.

✔ Include comparisons with baseline models and state-of-the-art, where applicable.

Limitations and future work:

✔ Include a discussion of the limitations of your approach and potential areas for future work.