Fine-Tuning Transformer Models on OPUS Books for German-French Translation

Varun Karwa (428443)
RPTU Kaiserslautern, Department of Computer Science

Note: *This report contains a project documentation and reflection on the portfolio task submitted for the lecture Engineering with Generative AI in WiSe 2024-25. This report is an original work and will be scrutinised for plagiarism and potential LLM use.*

1 Portfolio documentation

Compile a comprehensive documentation of your project, including all the project phases. You will need to explain every choice you made during the project and your thoughts about the results you get. You will introduce the results in suitable visualisation. Furthermore, you will need to explain which criteria you follow to build your prompts and how they affect the results.

Students write the entire documentation with sections, sub-sections, diagrams, etc in this section. Please write as comprehensively as possible. Head to the document 1_documentation.tex. You are free to use as many subsections as required. We will not provide a template for documentation.

1.1 Research Phase

The main purpose of this phase was to select benchmark Dataset and general purpose pretrained Model to be fine-tuned.

1. Dataset Selection

   I have chosen the Opus Books [1] dataset which contains 1,250,632 rows and has sentences in around 16 languages, also including German and French. It is a multilingual parallel corpus derived from translated books from a wide variety of genres, providing a diverse range of linguistic styles and vocabulary. This ensures accurate, fluent, and contextually meaningful translations, which helps improve model performance. It also contains simple and complex sentence patterns. It also contains various lengths of sentences(short and long, both), which makes the model more robust.

   More than 100 models have been fine-tuned or trained on this dataset, which satisfies the criteria of the benchmark dataset.

2. Model Selection

   The Meta/LLaMA 3.2-3B model is used for fine-tuning in this task. It is developed by Meta AI and part of LLaMA 3 series, designed for efficient, high performance language understanding and generation. The LLaMA model comes in 1B, 3B, 7B parameters configurations. This model has 3.21 billion parameters which exceeds the minimum requirement of 1 billion parameters.

It is a general-purpose language model, designed for a wide range of natural language processing tasks, such as text-generation, summarization and more. It has been pretrained on diverse-text, hence does not have randomly initialized weights, and it can be fine-tuned for translation tasks.

As it's size is less than 16GB it can be fine-tuned on colab.

1.2 Design Phase

1. Fine-Tuning Approach: LoRA

   Throughout this phase, I have explored various fine-tuning methods, such as Supervised Fine-Tuning, Parameter-Efficient Fine-Tuning(PEFT), and Full Fine-Tuning. Each technique requires different amount of resources and has individual impact on model performance. For example, full fine-tuning takes an excess amount of computational resources, while techniques like prompt engineering requires less in comparison. However, each method has different pros and cons.

   In this task, Quantized Low-Rank Adaptation(LoRA) [2] fine-tuning approach is used. It is a parameter-efficient method, which reduces memory and computational requirements by quantizing the weight to lower precision by 4 bit and also freezing the pre-trained model weights and updating only adapter layers. Since only a smaller number of parameters are updated, fine-tuning is faster compared to other methods.

   By freezing layers it preserves the original model's knowledge while getting adapted to translation task. Hence, it is ideal for fine-tuning large models on a Colab Notebook.

2. Dataset Split Ratio

   As we are using only 1000 pairs in Dataset A, there will be less sentence pairs for model A to be trained. Therefore, we have to select such ration such that there are enough pairs to be used for training and also for testing to know if model is predicting accurately or not.

   Hence, I have applied 70-30 ratio, reserving 700 pairs for training and remaining 300 pairs for evaluation.

3. Prompt for Generating Synthetic Data

   The Prompt for Generating German-French Sentences Pairs is carefully designed by a combination of multiple contexts and examples to ensure variation in sentence structure and meaning. This approach prevents the model from generating repetitive sentences and promotes a wide and diverse range of translations with more natural and various contexts. The model used to generate the dataset is Qwen-2.5-coder-32B-instruct [3] [4], which requires precise prompt to work otherwise accuracy could differ and irrelevant data might be created.

4. Evaluation Metrics: Bleu Score

   To evaluate models, the BLEU [5] metric is employed. For assessing text generating tasks like translations, it offers a consistent and repeatable metric. Higher scores indicate a perfect match between the generated text and reference material, while lower scores

indicate no overlap. The score runs from 0 to 100. It compares n no of sequences(n-gram) to compare the predicted translations with reference translations. The more n-grams matches, the higher the score.

1.3 Implementation Phase

In this phase, the implementation of all parts is in a Colab notebook.

- The notebook is divided into 13 parts including stretch goal also mentioned in the task description.
- All sections contains comments as required to explain logic embedded in code. Some parts are also taken from the exercises but updated according to the usage.
- Data Augmentation is implemented for generation of dataset B as per the task which is 2 time the original dataset A.
- The notebook also includes the visualization of the evaluation scores of each model, making it easier to compare which is better from all four.

1.4 Testing and Evaluation

The main purpose of this phase is to fine-tune the pre-trianed model three times using the opus dataset, synthetic dataset, and combined dataset sequentially and to evaluate them on test dataset A.

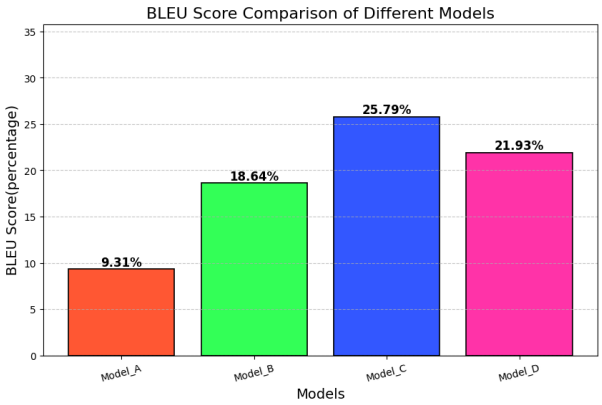

Figure 1: BLEU Scores of all models.

- The evaluation of models is done on the basis of the BLEU score1. During the model evaluation, the test dataset is tokenized and then German sentences incorporated with the prompt are given to the model to predict the French translation. The prompt has been kept simple to check if model can understand and translate the sentence and return french sentence without adding any jargon to it. But it can be seen in the evaluation output that sometimes the model returns English translation, or exact German sentence, or nothing at all, or mix it up.

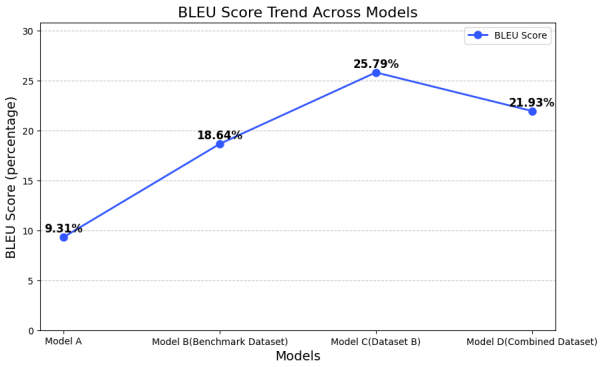

Figure 2: Trend across all models

- The comparison of different models concerning the BLEU score2 have clearly shown an improvement in the score as the models train with the appropriate result, except for Model D. Among all models, model C has achieved the highest score, while model A performed worst. Since model A is a general purpose model, not trained particularly for translations, it was expected to have a low score. Model B is fine-tuned on 700 German-French sentence pairs of opus books dataset, showed an improvement over model A, as fine-tuning enhanced its translations accuracy. In the case of Model C, it is fine-tuned on a synthetic dataset generated by the Qwen-2.3-coder-32B-instruct model, which consists of around 1400 German-French sentence pairs. So in comparison to model B it has been trained on twice number of sentences, however accuracy of model depends on the quality data generated. But to surprise, it has also outperformed model D. It was unexpected as model D has been trained on 2100 rows of data, so it was assumed that it would be the best model, indicating that more data alone does not guarantee better-performance-data quality plays vital role in achieving accurate translations.
- However, the current results are not guaranteed to be similar for every run. The synthetic dataset is generated by another LLM, since LLM do not always produce identical response for same prompt. As a result, if the training data change, the model's performance could also alter, leading to different BLEU scores with different models performing best amongst all. This highlights the importance of dataset consistency and quality control for fine-tuning models with synthetic data.

1.5 Interface for the translation(Stretch Goal)

During this phase, an interface has been built to query the model C which has the best BLEU Score as shown in figure 3

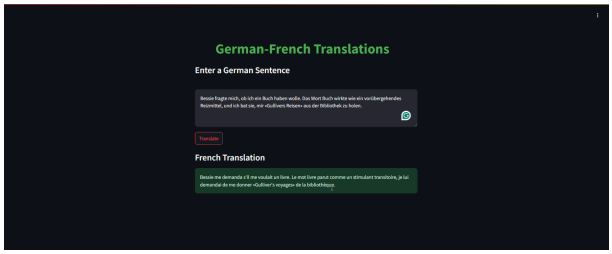

Figure 3: Translation Interface Image for Best Model.

2 Reflection

In 3-5 pages, 1500-2000 words

This section needs to be adjusted to align with the reflection requirements specified in the selected task.

Note: You should address all the questions from your selected task. Please list each question and provide your answers in the following enumeration.

1. What was the most interesting thing you learned while working on the portfolio? What aspects did you find interesting or surprising?

   Answer: The main purpose in enrolling this course was to get a fair understanding of the LLM during this project. I was introduced to many things which were new, such as synthetic data generation using dynamic prompt with examples from original dataset and using the generated data to fine-tune the pretrained model. This was the most interesting thing that I learned while working on the portfolio, as previously I was accustomed to training models on available datasets from different sources.

   I initially assumed that model fine-tuned on synthetic data (model C) would underperform compared to one fine-tuned on original dataset (model B). However, the BLEU scores revealed that the model C actually performed better than expected. One possible explanation for this is the balance of the dataset. Our evaluation indicates that the OPUS Books benchmark dataset might be unbalanced, which could have contributed to the lower BLUE score for model B. In contrast, the synthetic data appears to be more balanced but it cannot be guaranteed always as the dataset generated may vary everytime. When we combined the benchmark dataset with synthetic dataset and fine-tuned the pre-trained model(resulting in model D), interestingly, it's BLEU Score was lower than that of model C, which contributes in making our inference about benchmark dataset being unbalanced more acceptable.

2. Which part of the portfolio are you (most) proud of? Why? What were the challenges you faced, and how did you overcome them?

   Answer: The most difficult part of the portfolio exam was to generate the synthetic

data using Qwen 32B model. In the starting, I was giving a simple prompt "Generate 1400 pairs of German French Sentence containing different contexts" to the model for generation. But as LLM does not have short term memory, it can't remember after 3 sentences that, if it has produced similar sentence before and hence the dataset contains redundant sentences. To overcome this issue, I tried generating the dataset in batches by querying the model multiple times but still the problem was not solved and generation was not up to the expectation as the synthetic data contained repeated phrases, missing meaningful contexts and failed to maintain proper structure. To solve this, I though of passing different prompt to model per batch. The prompt contained randomly contexts selected from wide range of defined context and examples selected from original sentences for maintaining the structure.

The another problem which I faced was of making interface with the best model to query it. Due to the usage issue of google colab and model size, while running the interface, the colab was crashing due to ram overload. To overcome this, I tried writing the code for interface in separate file and calling it from streamlit, to make the interface running reducing the memory usage. Using localtunnel, to make the interface publicly accessible, ensuring seamless experience.

3. What adjustments to your design and implementation were necessary during the implementation phase? What would you change or do differently if you had to do the portfolio task a second time? What would be potential areas for future improvement?

   Answer: As I have already mentioned earlier, some adjustment was made in prompts for generating the synthetic data to improve it's quality and reduce redundancy. Another key adjustment was usage of QLora to fine-tune model instead of full fine-tuning due to restricted computational resources.

   If I can do the portfolio second time, I would like use full fine-tuning method on smaller model such as meta/LLaMA 1.3B, to compare it's effectivness against QLoRA fine-tuning. Also, I would like to fine-tune the model on benchmark dataset other than OPUS Books to see if the results are same, or the dataset makes the difference. Another one is to find the appropriate value of maximum new token for evaluation as this time model generates some garbage content in addition to french translation.

   The potential of improvement in future can be the use of more efficient models with high trainable parameters instead of Low-Rank Adaptation fine-tuning. This way we can avoid translations and make models more accurate. Also, I would like to make the model available in cloud to query it for translations to overcome the problem of size issues and restricted computational resources. And, training model to translate the sentences in languages which it has been trained yet.

4. Include a brief section on ethical considerations when using these models on language translation tasks. **Answer:** While using Large Language Models(LLM) for translations tasks, it is important to consider multiple ethical implications to ensure fairness, accuracy and responsible deployment.

   - Data Usability: The benchmark dataset used for fine-tuning does not belong to us, but to original authors. We are only using it for model optimization and not claiming

ownership. This thing should be done to ensure proper data handling and compliance with copyright regulations.

- Transparency in fine-tuning and Bias Awareness: It is important to understand how AI models process translations and whether they have any bias. As AI translation systems affect global communication, it is necessary to check whether they unintentionally or intentionally have biasness towards certain culture perspectives or ideology. Hence, fine-tuning process should be transparent, so that the generated translations are safe, reliable and free from bias that change real meaning.
- Explainability: It is one of the important factor in assessing AI translations, the ability to understand how and why the model translate the sentences. If model makes mistakes in translations, we need to identify the cause, such imbalanced dataset, limitaions in training, or issues in fine-tuning. While using the fine-tuned model in real-world applications, it is also important that users understand how it works and how it can be improved to make translations more accurate.

5. From the lecture/course including guest lectures, what topic excited you the most? Why? What would you like to learn more about and why?

   Answer:

   - The lecture with RAG excited me the most. I find it fascinating because it allows models to search for updated information instead of relying on pre-trained knowledge. I want to explore this field in respect with LLM, how they enhances the model results and deep understanding of them. Even now, I'm using RAG in my guided research with LLM to extract data analytics pipeline from research papers. Using RAG helped me reducing the output generation by 10 times, by retrieving relevant context from papers and passing to LLM rather than whole paper.
   - Another lecture which excited me was with the Prompt, where they taught how different prompt techniques can be incorporated. I want to know hoe prompts can improve models without the need of full or partial fine-tuning method. Since training LLM requires excess amount of computational resources, understanding how to design effective prompt can help enhancing results with minimal computing resources.

6. How did you find working with DIFY platform during the course work? Would you recommend using DIFY in learning Generative AI technologies and why? What is the best start for learning Generative AI either by Python code or No-code platforms and why?

   Answer: During the first assignment, I used the DIFY platform. It was an interesting and insightful experience. As it has user-friendly, no-code interface, makes easier to test and experiment different models and prompt. It was also easy to make working interface and query the model to produce the results. Also the drag and drop functionality and pre-built tools simplified the process to make AI projects, and evaluate model in different test case scenarios.

   DIFY can be recommended to beginners without any knowledge of coding. It allows users to directly focus on AI concepts instead of indulging in complex coding.

The best way to start learning Generative AI depends on the learner's ability to code. For those just beginning, no-code platforms such as DIFY, hugging face spaces are excellent options. These tool enable users to explore models, adjust prompts and analyze results without any programming knowledge. While, those who want to understand the working of AI models can deep dive with python-based learning. They can use libraries like Hugging face transformer, Tensorflows, etc to train and fine-tune their own models, optimize performance. It provides more control and is important for those who want to build more complex AI systems.

7. How did you find the assignments and exercises in the course and how they help you in portfolio exam?

   Answer: The course assignments and exercises were complex, but implementing them gave a great understanding of various topics. Such as exercise 3 is some-what similar to the portfolio task. Going through exercise 3 helped me understand how different fine-tuning techniques like PEFT, Full fine-tuning works on making model efficient. Other assignment and exercised taught me important skills like understandign LLM, prompt engineering and preparing data, which were important aspects for portfolio task.

References

[1] 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).

[2] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[3] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[4] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

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