Large Language Model Refinement for Neural Machine Translation

Peter Valverde[1]*

[1]School of Information, The University of California Berkeley

Abstract

Conventional machine translation methods utilize encoder-decoder model architectures to perform neural-powered machine translations (NMT). These models take an input sequence in one language, encode the meaning of the text into a language-agnostic number space, and then generate a new sequence in the target language by decoding the encoded meaning into natural language text. This approach has led to significant improvements in the fluency and adequacy of machine translation models, however we believe that this architecture leaves room for improvement. Due to the recurrent, one-pass limitation of the NMT architecture, the final translations of these models are generated sequentially without the bidirectional context of the entire generated translation (both the prior-generated tokens as well as the subsequent tokens). In this work, we propose an architecture leveraging fine-tuned large language models (LLM) to refine the output translation of the NMT model. Importantly, this enables a final translation that factors in the source text as well as the NTM first-pass translation wherein our fine-tuned LLM can focus purely on the translation refinement (fluency) while allowing the NTM model to handle the initial leg-work (adequacy).

Introduction

The neural network encoder-decoder framework has been applied to several natural language processing tasks, often to great success. Namely, with regard to machine translation this architecture has yielded impressive results (Sutskever et al., 2014). In this context, these models work by encoding the text in the source language, extracting the meaning and representing it numerically, before passing these encodings to a decoder which reads this context and uses it to generate text in the target language. This decoding process is performed sequentially meaning that the model can only leverage the encoded context from the source text and prior-generated tokens in calculations to generate each subsequent token (Vaswani et al., 2017).

Due to this inherent limitation, the model may incorectly generate certain tokens and because there is no process in place to revise the generated translation, these errors will persist through to the final generated output. Once the translation is completely generated, we can intuitively see where the model may have gone wrong, identifying the token(s) that were translated incorrectly, but without the context of the full translated sequence, these errors would likely be harder to spot. This dilemma spawns the idea of generated output refinement which allows the model to evaluate and make changes to the first-pass-generated sequence before producing the final translation output (Chen et al., 2022).

*Corresponding author: pverde1@berkeley.edu
Received: December 8, 2024

Prior Works

Many have tackled the task of refinement with regard to generated machine translation outputs. One such approach involves neural system combination which takes three inputs: the source text as well as a NMT and statistical machine translation of said text, and encodes all of these sequences into an encoder-decoder model to generate a final translation. This approach allows the model to reference not just the source text but also two independently generated translations when creating the final output translation (Zhou et al., 2017). Another approach leverages synchronous refinement to evaluate the prior-generated token simultaneously while generating each subsequent token so that the generated translation is reviewed and potentially tweaked over several passes; this enables the model to make changes to the generated text before generating a subsequent token, which may help improve the accuracy of the newly generated token as well (Chen et al., 2022). Finally, the refinement approach which most motivated the work we propose today: a deliberation network which allows a second-pass over a fully generated translation in addition to the source text such that the second decoder can access the full context of the source and a proposed translation while generating the final sequence (Xia et al., 2017).

Our Approach

All of these works as well as many others are trying to solve the same key issue that presents in traditional NMT models. Intuitively, a single-pass generation method that fails to incorporate the context

of the entire translation from end-to-end leave potential translation fluency on the table. While all of these refinement approaches have yielded positive results, we believe that the most recent developments in large language models, fine-tuning, and parameter efficient approaches can be leveraged to further contribute to the literature around refined machine translation.

For our work, we propose a new method of machine translation refinement: fine-tuned LLM refinement. Inspired by the various methods of synchronous and asynchronous refinement, we determined to attempt our own approach of asynchronous refinement by fine tuning various large language models to take the source text as well as a NMT single-pass translation as inputs to our fine tuning model. We determined to enact this approach on a sample of the WMT19 German-to-English dataset, one of the most cited, publicly-available datasets. While our approach has shown some weaknesses, we still believe that further work into our approach could have the potential for state-of-the-art improvements in the field of machine translation.

Background

Large Language Models

BACKGROUND BASICS ON LLMS

Intuition

When machine translation models are scores, multiple sample texts from the source language are passed into the model to be translated. These translations are then scored for their quality (fluency and adequacy) either by professional human translators, standardized metrics (BLEU, ROGUE, etc), or both. These evaluation methods take in both the source language as well as the translation when evaluating the quality of the translation. Intuitively, human translators may read a text and identify one or multiple words in the translation that may dock the score of the translation.These incorrect words may either negatively impact the translation's fluency (the words to not make sense in the target language) and/or adequacy (the words do not properly convey the meaning of the source text. Following from this methodology of evaluation, our approach was to mimic a human translator's evaluation of NMT by leveraging the most recent developments in AI through LLMs.

Our Application

Our approach is to train an LLM to learn patterns in the shortcomings of traditional NMT translations of texts. With these learnings in tow, the fine-tuned LLM could then take a source text as well as a NMT translation and apply the learned corrections to the NMT translation with the aim of refining the NMT output into a higher quality translation. As mentioned above, the capabilities of LLM in a variety of tasks is already impressive, with our approach we hop to further enhance the LLMs abilities by specifically training the model to mimic the approach a human-translation evaluator takes: understand the source text, read the proposed translation, identify and correct the parts of the translation that need improvement.

Methodology

Source Data Set

For our work, we wanted to be sure we were using a large, tried and true dataset for our model training. Because the novelty of our work is entirely focused on the application of LLMs for NMT output refinement, and not regarding apply an existing technique to a new domain or larger training set, we wanted to ensure the use of an industry standard dataset. With this in mind, we utilized the WMT19 German-to-English dataset. This data features over 34 million records of German sentences and their corresponding English translations, and while we leveraged no where near that many records for our work, the ability to our approach in future works is an enticing aspect of this dataset.

Generating NMT Translations

As the critical input of our proposed model, it was important to obtain high-quality machine translations. Our search for publicly available translations, including source text, NTM translation, and a professional human translation, yielded no viable datasets, so we determined to generate a dataset containing just the source text and the human translation and undertake the NMT translation aspect ourselves. For this, we wanted to ensure that we utilized a highly accurate and broadly accessible NMT model so as to ensure our baseline translations were of high quality and that future work could be reproduced and improved upon without being hindered by the availability of our NMT model. We decided to utilize Meta's M2M100 NMT Model to create our single-pass NMT translations. This model is readily accessible, highly regarded, memory conscious, and has the added benefit of being able to directly translate between 100 languages without using any intermediary languages. This feature enables our methods to

be applied directly to many more language combinations in future work. To create NMT translations for our model input, we passed our 100,000 german text examples into the M2M100 and recorded the English translation outputs.

Fine-Tuning Llama Models

There are a number of approaches to fine-tuning LLMs, but for our work we leveraged the Hugging Face Supervised Fine-Tuning Trainer. For this we had to format the fine-tuning input data in such a way that the LLM could differentiate between the input prompt and our desired output translation. Specifically, we prompted the LLM to "fix" the German to English translation, passing in both the source german text and the NMT english translation as well as the target translation from the WMT database. In order to train the model and differentiate between our input prompt and the desired LLM output we utilized the <|im_start|> and <|im_end|> tags which is a conventional formatting step prior to fine-tuning LLMs.

To fine-tune our LLMs in a memory efficient and timely manner, we leveraged the Low-Rank Adaptation of Large Language Models (LoRA) training technique (Hu et al., 2021). To prevent overfitting, and enabled by LoRA's efficiency we trained each of our model versions on only half of an epoch through our training data.

Experiments

We perform our NMT translation refinement experiments by training on a number of Meta Llama LLMs and leveraging different training techniques. We chose to use Meta's Llama models for our base LLMs as they are some of the best performing, open source models. Additionally there are several model variants available from Meta which enables us to specifically select the LLMs that best fit out needs. The final determination in selecting Llama models for our research was that having a several models within a standardized framework allowed us to run multiple experiments easily substituting in different models with minimal reformatting or dependency changes.

Llama3.2 1B Without Source Text

The first model we trained was a fine-tuned version of the Llama3.2 1B parameter LLM. We chose to start with this smaller model as we figured that given the M2M100 Model utilizes only 5̃00M parameters, we may be able to get away with the smallest available llama model which still has nearly twice as many parameters. For this initial model, we fine-tuned using a prompt to "rewrite" the NMT generated translation and set the WMT translations as the desired LLM assistant output. We initially prompted to "fix the translation", but found that this led to poorer refined translations. This is likely because for this model we did not pass in the German source text, so the LLM was when asked to correct a translation without the source translation to go off.

Llama3.1 8B Without Source Text

After training and evaluating the aforementioned Llama 3.2 1B model, we decided to implement the same training procedure again, but this time utilizing the Llama 3.1 8B model as our base LLM. While the Llama 3.1 models are older than the 3.2 models, we figured that the higher parameter model may still outperform the smaller yet newer 3.2 model for our purposes. What we found with this experiment is that the Llama 3.1 8B model did indeed outperform the Llama 3.2 1B model which helped guide our model selection in the proceeding experiments.

Llama3.1 8B With Source Text

As mentioned above, our first two models did not utilize the German source text in their fine tuning approach. While at first our idea was to focus on refining purely off of the NMT outputs, we found that our refined translations were often losing adequacy relative to the NMT translation. Without the source text, these models were missing crucial information for generating the highest quality translations. This motivated us to perform a second round of fine tuning wherein we prompted the LLM to "fix" the German to English translation, passing in the German source text as an additional argument in conjunction with the NMT translation and WMT translation as before. This new prompt and input allowed our fine-tuned model to learn off of a more complete picture of the translation process and did indeed lead to improved results over the translation-only models described above.

Results

Evaluation Metric

When comparing all of the available evaluation metrics for natural language processing tasks the one that first stands out is almost always the BLEU score. A tried and true metrics, the BLEU score uses n-gram similarity between your generated and target sequences in order to evaluate how well the generated sequence matches the target. While BLEU has been widely used for years, we did not feel that it was

Table 1: *Model Performance: BLEURT Scores*

Model Version		
Model	With Source	BLEURT
Llama3.2 1B	False	-0.51986
Llama3.1 8B	False	-0.35967
Llama3.1 8B	True	-0.22415
M2M100	True	0.23266

best suited for our purposes of evaluating the translation quality of refined NMT outputs. It is because of this that we settled on using BLEURT as our objective evaluation metric. BLEURT is a trained model that encodes both the generated input and target sequences in order to better evaluate the consistency in meaning between the two sentences (Sellam et al., 2020). Leveraging BERT, BLEURT is able to evaluate the semantic similarity between the sequences rather than just the tokens in each, which is why it is the ideal evaluation tool for this research. The more positive the BLEURT score, the more semantically similar the inputs are to the target.

Model Performance

As discussed above, we chose BLEURT as the objective metric to evaluate our fine-tuned models. For this evaluation we held aside 1500 records from the WMT19 German-to-English Validation Dataset so that we could have an unbiased and representative look at how our models fair. We ran the German source text from each of these records through the M2M100 model to get our intermediary translations to serve as input to our fine-tuned LLMs as well as a baseline dataset against which we could measure our models. We then passed the NMT translations through each of our three models and recorded the refined translation outputs. We mirrored the corresponding prompt that was use to fine tune each model for the inference invocation to ensure the best results possible.

As shown in Table 1, none of our three fine-tuned LLMs outperformed the translation quality of the M2M100 model. Our best performing model was the Fine-Tuned Llama 3.1 8B with Source German Text with a BLEURT score of -0.22415 on our 1500 record validation set. While we were not able to beat the state-of-the-art with our refinement methodology, we did see considerable improvement made with each subsequent model we implemented. This indicates that our perhaps our proposed approach is on the right track, as we are moving closer and closer to the baseline with each model iteration.

Beyond the Metrics

What is so challenging and intriguing about natural laguage processing is that the numbers can only tell you so much. In this section, we take a step back from the numbers and see what is really happening in each of our models. WIth our V1 models which excluded the source text from the fine tuing process, we see many 'refined' translations completely lose their original meaning. Without the context of the source text, the fine-tuned LLM would take it upon itself to largely change the meaning of the sentence, or at times just carry on adding new sentences to the translation. These problems somewhat disappeared when we passed the source text into the fine-tuning process such as in our final model. While this model sometimes still made large changes to the meaning of the source text, all in all it did a better job of keeping true. These creative additions were not due to the temperature of the LLM as repeated invocations of the model would often lead to the same refined output, which we would not expect to see if the temperature parameter was responsible for the variability of output.

Discussion

While we were certainly disappointed that our approach did not beat the state-of-the-art in this work, we are still happy to present the findings we uncovered. We believe that through additional model iterations, fine-tuned LLM refinement may still show to be a powerful tool for improving the adequacy and fluency of machine translation outputs. Perhaps future work with larger datasets, further refined prompting, different source LLMs, or novel fine-tuning approaches may yield more significant results. The novelty of our approach just means that there is more work that can be done, and surely further enhancements can be made. While our goal was to beat out the conventional M2M100 model, what we have seen here is that this NMT model is already incredibly refined for translation tasks and beating its performance will have to be a goal achieved by future work.

Conclusion

In this paper, we proposed a new approach at NMT output refinement utilizing fine-tuned LLMs. We added to the corpus of work aimed at improving upon the translation quality of neural machine translation, and while our methodology has not shown to beat the state-of-the-art, we maintain that the exploration of LLMs as a tool for output refinement is a worthwhile endeavor. In our work, we were able

to achieve continuous, substantial improvement with each subsequent model version, which is why we feel additional work in the field may yield even better results.

References

[1] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In Proceedings of NIPS 2014.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc.

[3] Kehai Chen, Masao Utiyama, Eiichiro Sumita, Rui Wang, and Min Zhang. 2022. ynchronus Refinement for Neural Machine Translation. Findings of the Association for Computational Linguistics: ACL 2022, pages 2986 - 2996.

[4] Long Zhou, Wenpeng Hu, Jiajun Zhang, and Chengqing Zong. 2017 Neural system combination for machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 378–384

[5] Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In Advances in Neural Information Processing Systems 30, pages 1784–1794.

[6] Edward Hu, Yelong Shen, Philip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models.

[7] Thibault Sellam, Dipanjan Das, Ankur P. Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation