Exploring the Effectiveness of LLM Domain Adaptation
for Business IT Machine Translation

Johannes Eschbach-Dymanus   Frank Essenberger   Bianka Buschbeck   Miriam Exel
SAP SE
Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany
`firstname.lastname@sap.com`

Abstract

In this paper, we study the translation abilities of Large Language Models (LLMs) for business IT texts. We are strongly interested in domain adaptation of translation systems, which is essential for accurate and lexically appropriate translation of such texts. Among the open-source models evaluated in a zero- and few-shot setting, we find Llama-2 13B the most promising for domain-specific translation fine-tuning. We investigate the full range of adaptation techniques for LLMs: from prompting, over parameter-efficient fine-tuning to full fine-tuning, and compare to classic neural machine translation (MT) models trained internally at SAP. We provide guidance how to use training budget most effectively for different fine-tuning approaches. We observe that while LLMs can translate on-par with SAP's MT models on general domain data, it is difficult to close the gap on SAP's domain-specific data, even with extensive training and carefully curated data.

1   Introduction

With swift improvement and recent successes of Large Language Models (LLMs), it has become imperative for companies to measure their productive NLP systems against such new models. In the rapidly evolving field of NLP, incorporating the state-of-the-art models could unlock new capabilities for one's product and improve performance. On the other hand, switching to LLM-based systems should not be done merely to appeal to public hype, but should be a thoroughly evaluated choice.

With this in mind, we set out to investigate whether LLMs can be easily utilized to outperform and ultimately supersede the current machine translation systems employed by SAP (Buschbeck et al., 2022). They are based on a traditional neural machine translation architecture trained on a multitude of data sources including the contents of the company-internal translation memories and is therefore optimized for SAP's domain of interest, which we call *Business IT* here. While previous research has shown that LLMs make good translators (Hendy et al., 2023; Zhang et al., 2023; Zhu et al., 2023), it is not yet well explored whether they can effectively adapt to domain-specific translation intricacies and outscore a smaller model that has been trained from scratch within the domain.

In particular, our interest lays in whether comparably smaller sized open-source LLMs can be fine-tuned to this end. This interest is motivated by certain drawbacks of using large proprietary models such as OpenAI's GPT-4 (OpenAI et al., 2024) out-of-the-box. Potential data privacy concerns, slower inference and higher monetary costs (provided sufficient throughput) are some of the reasons.

In addition, fine-tuning (open-source) models offers some more benefits. Fine-tuning addresses challenges such as hallucinations and overgeneration, commonly associated with the LLMs' innate generative nature. By channelling the LLMs' focus towards translation through downstreaming, it becomes possible to regulate and control these unwanted generative tendencies, resulting in a more precise and tailored output for the intended domain-specific application.

While there is a general argument to be made that fine-tuning an LLM on parallel data would im-

prove its translation quality, a much more crucial one can be made regarding domain-specific translation. In domain-specific translation, vocabulary and translation patterns might differ substantially from its general purpose counterpart. For instance, UI strings contained in our domain-specific data such as 'Item masters in catalog', 'Number in the Total field' or 'PENDING The item has not yet been sent.' do not only differ from general domain texts in terms of vocabulary, but can also deviate substantially in syntax.

In our experiments, we first estimate the translation capacities of various models through prompting to identify the most promising one to fine-tune. Concretely, Llama-2 13B appears to offer the best model size to performance trade-off. We then fine-tune the model with both full parameter and parameter-efficient tuning and conduct various ablation studies.

Overall, we arrive at the following conclusions:

1. It requires large amounts of domain-specific data to have a fine-tuned Llama-2 13B approach the performance of a smaller but dedicated translation system.

2. When fine-tuning Llama-2 13B with domain-specific translation data, we do not observe noteworthy catastrophic forgetting of general domain translation capacity.

3. For domain-specific translation, low-rank adaptation (Hu et al., 2021) cannot compete with full fine-tuning as it fails to internalize domain-specific phenomena in the limited parameters available. In order to increase model fit, the adapter rank needs to be increased to magnitudes where the tuning becomes no longer parameter-efficient. Additionally, we observe that quantization significantly hinders model fit and consequently in-domain performance as well.

4. If nonetheless parameter efficient fine-tuning is conducted, one should favour an increase in training data over an increase of training epochs. For full fine-tuning, however, training for multiple epochs provides a notable benefit. In fact, if the number of training iterations are to be kept a constant, one should favour an increase of epochs over more training data; naturally, under diminishing returns.

2 Related Work

In their paper, Brown et al. (2020) presented the performance of GPT-3 and evaluated the translation ability of their LLM. They found that general-purpose LLMs benefit from having examples in the prompt (few-shot prompting) to guide the model towards a specific task. Undoubtedly, adding relevant examples via few-shot prompting or retrieval augmented generation can improve translation performance. However, both Alves et al. (2023) and Li et al. (2023) observe that translation fine-tuning outperforms few-shot prompting when provided with only few thousands of training samples. Xu et al. (2023) achieve state-of-the-art translation performance with help of a two-stage training mechanism, where in the first stage, the model is further pre-trained on billions of tokens of monolingual data of various languages to shift the model to a more multilingually balanced state; away from its dominantly English pre-trained state. Only then, the model is fine-tuned with limited parallel data. While the resulting performance is astonishing, the first training stage is computationally expensive. Even with adequate GPU resources available, the proposed setup does not necessarily work as effective in low-resource domain translation, where the parallel data does not align well with the monolingual data used in the first stage. Üstün et al. (2024) recently presented the Aya model which uses a more balanced distribution of multilingual data in the pre-training. Although the approach seems promising, our first preliminary investigations do not show substantially higher translation performance of the Aya model compared to previous LLMs.

3 Datasets & Evaluation

In this paper, we mainly focus on the high-resource language pair *English → French*. The more resources a language pair has, the more LLMs should be able to leverage from their pre-training, making it easier and quicker to downstream them for the translation task. In addition, we also investigate the performance on the low-resource language pair *English → Slovak* and on *English → Japanese*, which is known for its complexity, in section 5.7.

For few-shot example retrieval, fine-tuning and testing, we use well-curated parallel SAP-internal data. It is composed of large amounts of software user interface (UI) strings, user assistance (UA) texts, but also training materials, corporate content and marketing texts. The models are tested not

only on a test set of 2000 segments consisting of domain-specific UI strings and UA texts, but also on general-domain data, i.e. the FLORES (Goyal et al., 2022) test set. Even though they stem from the same domain (SAP), training and in-domain test data are not merely divided randomly, but instead feature both temporal and distributional shifts. This in turn allows a more realistic performance evaluation (Søgaard et al., 2021).

We want to clarify that SAP's MT systems are trained using large corpora comprised of millions of parallel sentences, and this training is performed over many epochs. Since fine-tuning LLMs on a similar scale would entail considerable computational costs, we conduct our fine-tuning experiments with fewer but gradually incremented quantities of data to map respective improvements in translation quality. Furthermore, while SAP's MT system evaluated in this study has been trained to excel in translating texts from the SAP domain, it has not been fine-tuned to UI and UA texts specifically, and obviously the test data is unseen.

We evaluate the performance with both BLEU (Papineni et al., 2002) and COMET[1](Rei et al., 2022). While COMET is more robust and correlates better with human annotators, the n-gram based BLEU score nonetheless has its use when evaluating domain-specific translations. Specifically, it captures lexical agreement with references which indicates the correct use of terminology and writing style for the domain.

4 Prompting

To establish a baseline for the translation fine-tuning it is natural to start with simple prompt experiments. These experiments are relatively straightforward to conduct since large commercial models like GPT-4 are offered as services. Furthermore, hosting open-source models for inference requires fewer resources compared to tuning them. The motivation driving this inquiry is twofold: first, to establish a baseline for the performance of open-source models; and second, to evaluate the inherent capabilities of GPT-4, a proprietary cutting-edge language model in its "out-of-the-box" state.

Given the nature of the task, we restrict ourselves to a selection of models that are intended for multi-lingual usage:

1. **GPT-4**: GPT-4 serves as a benchmark for the

state-of-the-art in natural language processing, boasting superior language understanding and generation capabilities (OpenAI et al., 2024). Hendy et al. (2023) have also demonstrated that it shows remarkable performance in the translation task.

2. **Llama-2 (7B, 13B, 70B)**: The Llama-2 family of models has been shown to achieve great performance in various tasks across various languages and has a commercially usable licence.

3. **BLOOM 7B**: The BLOOM model family has been released in 2022 and was trained on well documented high-quality data (Laurençon et al., 2022) encompassing 46 natural languages.

4. **Falcon (7B, 40B)**: As with BLOOM, the Falcon family of models, released in 2023, is of special interest due to its balanced, curated and, most importantly, well documented multilingual training data (Penedo et al., 2023).

```
English: Receive supplier invoice
French:
```

Figure 1: Translation Prompt

The simple prompt shown in figure 1 is used for prompting and fine-tuning experiments throughout the paper. While it is known that optimized and more verbose prompts can improve results, we refrain from *prompt engineering* for two reasons. Firstly, while engineering prompts is fairly cheap when optimizing a zero-shot setting, it would require repeated trainings for each and every prompt to measure its performance in a fine-tuning setting; an endeavour that is too costly. In our experiments, we expect the model to adapt to any prompt during fine-tuning. Susceptibility to prompt design would prove a major obstacle for fine-tuning LLMs. Secondly, a short and concise prompt is preferable as it leaves more context length available for the actual translation pairs.

Figure 2 displays the BLEU and COMET scores of the models in the zero-shot setup. For open-source models, we establish a comparison under equal resource conditions, i.e. given four NVIDIA A10G[2]. Models that are too large for full precision inference are tested with 8-bit quantization instead.

As expected, SAP's MT model performs best by a large margin on the domain-specific data. The superior BLEU score, in particular, indicates the cor-
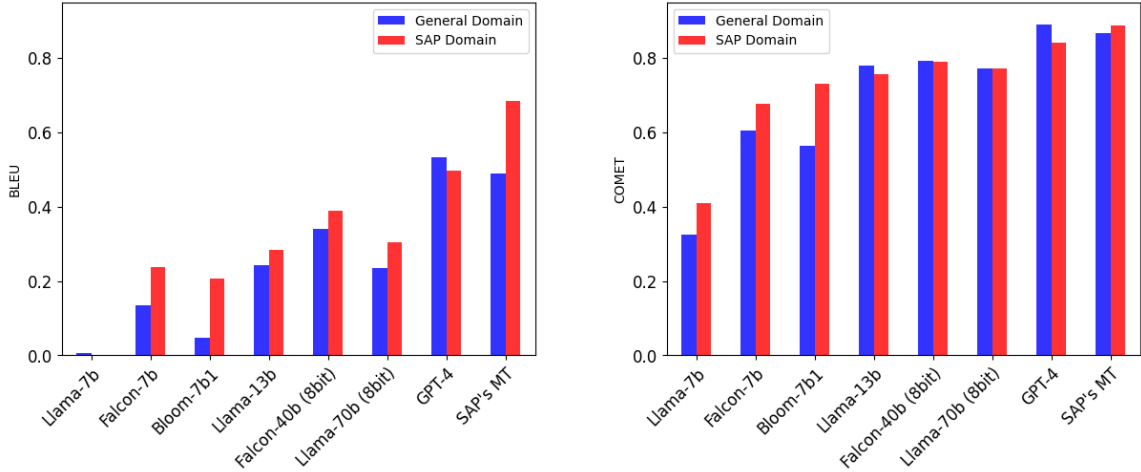

Figure 2: Zero-shot prompting performance

rect usage of in-domain vocabulary, rather than just semantically similar phrases. For general-domain data (FLORES), however, GPT-4 is capable of outperforming SAP's domain-specific MT system out of the box. Keep in mind that this MT model is not optimized for general-domain translation, and it is unknown whether the publicly available dataset was included in GPT-4's training data.

Provided with the simple prompt, all open-source models but Llama-2 7B are able to translate texts consistently[3], albeit not necessarily correctly. Naturally, it is difficult to determine whether the weaker model performance is the result of shortcomings in its translation quality, or merely a misinterpretation or mishandling of the prompt.

It might seem unintuitive, but some open-source models perform better on domain-specific data than on general-domain data. This can be explained by the large percentage of UI segments contained in the SAP test set. While such strings contain lexical intricacies, they are generally short and syntactically simple, which makes them easier to translate for the smaller models.

As a natural next step, we investigate whether few-shot prompting could narrow the gap for the SAP domain-specific translations. We focus our experiments on GPT-4, serving as an upper LLM benchmark, and Llama-2 13B, which offered the best trade-off between model size and translation performance in the zero-shot experiments. In order to construct our few-shot prompts, we first encode the English source segments of the domain-specific parallel training data (section 3) with the sentence-BERT model *all-MiniLM-L6-v2* by Reimers and

Gurevych (2019). Then, for each English segment to translate, we retrieve the five translation pairs that have the highest cosine similarity to the English source. These pairs are then arranged as completed prompts and placed before the final segment set for translation, with each pair separated by an empty line.

For the Llama-2 13B model we had to conduct postprocessing of the output due to overgeneration issues. In particular, it continued generating English-French sentence pairs beyond the completed prompt. To deal with this, we simply truncated the generation after the first line break. The results of the few-shot experiments are displayed in table 1. While both models improve in performance of in-domain translation, a gap to the domain-specific MT model still remains. However, it is worth noting that the open-source Llama-2 model benefits more substantially from the examples, promising potential that may be even better leveraged through fine-tuning. We also find that providing domain-specific examples for general-domain translation is detrimental to the models' performance.

While few-shot prompting can provide a basic understanding of the domain in question, it may not fully capture all the domain-specific nuances. This is particularly challenging when the domain is highly specialized, as selecting the appropriate domain-specific vocabulary and creating accurate examples can be difficult. Additionally, including multiple examples in the prompt increases the token count, which can lead to higher computational costs and longer inference times. Fine-tuning, on the other hand, would allow the model to learn

---

[3] generating output in French and not English

Model	SAP domain	General domain
Llama-2 0-Shot	0.757 / 0.283	0.780 / 0.243
Llama-2 5-Shot	0.843 / 0.503	0.767 / 0.310
GPT-4 0-Shot	0.842 / 0.498	0.891 / 0.532
GPT-4 5-Shot	0.866 / 0.560	0.883 / 0.518
SAP's MT	0.888 / 0.686	0.867 / 0.490

Table 1: Few-shot results with COMET (first value) and BLEU (second value) for Llama-2 13B and GPT-4. For comparison SAP's MT is added.

from a much larger pool of examples, potentially leading to better adaptation to the specific-domain requirements.

5   Fine-Tuning

While the performance of open-source LLMs is promising, it falls short compared to both a domain-specific neural machine translation model as frequently used in production and more advanced models like GPT-4. A promising pathway to improve the task-specific performance of open-source LLMs is to further fine-tune them. Given the substantial gap in performance on the in-domain data to SAP's MT model and even GPT-4, we have a strong incentive to investigate how well LLM translation can adapt to a specific domain through fine-tuning. To do so, we experiment with three different fine-tuning setups:

1. **LoRA**: With low rank adaptation (Hu et al., 2021), the pre-trained model weights are frozen while trainable rank decomposition matrices are injected on top of the frozen weight matrices. As the decomposition matrices are the only ones fine-tuned and contain magnitudes less parameters, model downstreaming becomes faster and more GPU efficient.

2. **QLoRA**: The fine-tuning approach proposed by Dettmers et al. (2023) quantizes the pre-trained model during training and only keeps the trainable LoRA adapter weights in standard precision. This method reduces the memory requirements of fine-tuning which in turn, depending on the available GPUs and model size, can allow data-parallel training rather than model-distributed one, cutting training time short by a multitude.

3. **Full fine-tuning**: While Hu et al. (2021) and Dettmers et al. (2023) show that the proposed parameter efficient fine-tuning approaches perform on-par with full fine-tuning, other researchers applying them could not always con-

firm such observations (Sun et al., 2023; Chen et al., 2022). Consequently, we also conduct full fine-tuning to establish an upper bound.

QLoRA is of special interest, as the greatly reduced GPU footprint allows cost-efficient training. In addition, quantization could also help reduce the cost during inference and recent development in dynamic adaptation (Babakniya et al., 2023) make QLoRA even more tempting. A main interest of our experiments is therefore an evaluation of QLoRA against full fine-tuning for domain-specific translation. Then, ablation studies are conducted that investigate shortcomings of QLoRA opposed to LoRA without quantization.

5.1   Fine-Tuning Setup

We use the 13 billion parameter version of Llama-2 for all fine-tuning experiments. For one, the model is capable of translation in a zero-shot prompt setup, which certifies that there is sufficient pre-trained knowledge to leverage through fine-tuning and allows a comparison to a sensible baseline. Secondly, the model is comparably lightweight, which allows full fine-tuning on as few as four NVIDIA A10G[4]. We use standard libraries to perform the fine-tuning, namely Huggingface's trainer interface[5] and bit-sandbytes[6] for quantization. We use the training data presented in section 3 and vary the amount of training segments in the experiments and train for 3 epochs.

Measuring performance not only in the domain-specific but also on general domain data allows us to investigate the effect the domain-specific translation tuning has on the model's translation performance in general. On the one hand, we expect an increase in general domain translation performance, as the model is downstreamed to translate only. On the other hand, increasing the model fit to specific data could also induce catastrophic forgetting and consequently cause the general domain performance to deteriorate.

For (Q)LoRA training, we set the rank to $r = 8$ and the scaling factor to $\alpha = 16$ unless otherwise specified. We use 8-bit quantization for QLoRA, as its 4-bit counterpart did not yield satisfying results in preliminary experiments. For both (Q)LoRA and full fine-tuning, we observed good convergence

---

[4]Using paged optimizers as discussed in Dettmers et al. (2023)
[5]https://huggingface.co/docs/transformers/main_classes/trainer
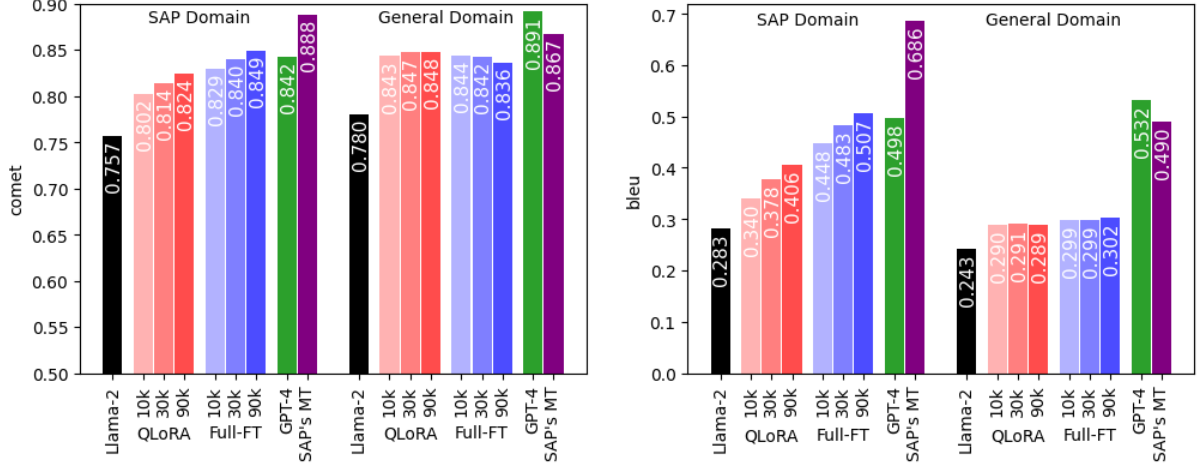[6]https://github.com/TimDettmers/bitsandbytes

Figure 3: Model performance measured in COMET (left) and BLEU (right). The Llama-2 models have been fine-tuned for 3 epochs with 10 to 90 thousand parallel segments.

behaviour with a learning rate of 2e-5. Other than heuristically searching for a functional learning rate, we did not search any further hyperparameters. After all, one key advantage of (Q)LoRA over full fine-tuning is that it is not as sensitive to hyperparameters; an advantage we do not want to offset by expending valuable resources to optimize full-fine-tuning.

5.2 Results

The results in figure 3 demonstrate how effective LLMs can learn from very limited training data. Even a small training set of only 10k sentence pairs drastically improves performance over the zero-shot baseline. This initial boost of performance, compared to zero-shot, is most likely due to the model quickly adjusting to the prompt and translation task in general. By increasing the training data we can further improve the model's performance, albeit with diminishing returns. At 90k training samples, the fine-tuned model surpasses GPT-4 performance on the domain-specific test sets. This demonstrates that fine-tuning of a smaller open-source model can close the performance gap to large proprietary models out of the box. With limited training data, however, Llama-2 cannot be easily downstreamed to beat the parameter efficient SAP translation system. Further investigations into the amount of training data required to match SAP's MT performance are conducted in section 5.6.

Despite training only on domain-specific SAP data, the model also shows improvements in general domain translation performance. While this is unsurprising, given that the model is downstreamed

for translation, it is nonetheless remarkable that there is no apparent catastrophic forgetting occurring when fine-tuning with the above quantities of training data. When full fine-tuning, we begin to see a slight degradation of general domain performance from 30k samples upwards. However, the performance is still substantially better than the model's zero-shot one. This general robustness also stands in contrast with few-shot prompting, where the addition of domain specific examples deteriorates general domain performance. In a way, one could argue that the few-shot examples much more aggressively urge the model to translate in the domain-specific style while fine-tuning only provides the models with the additional knowledge to translate appropriately, if necessary.

In general, we observe that full fine-tuning is superior to QLoRA tuning. Most notably, however, is that the full fine-tuned model displays a much larger improvement in BLEU scores on domain-specific data than its QLoRA counter-part. Since BLEU is a token-based metric, we conclude that full fine-tuning allows the model to internalize the lexical intricacies of the domain. This is crucial for translation use cases in specific domains, also at SAP, as the translations must be consistent with established terminology. With less trainable parameters available, QLoRA is less capable of internalising these lexical differences.

While full fine-tuning is undoubtedly the superior choice, it comes with increased computational costs. With the GPU setup discussed above, the QLoRA training could be conducted in a data distributed manner, while the full fine-tuning required
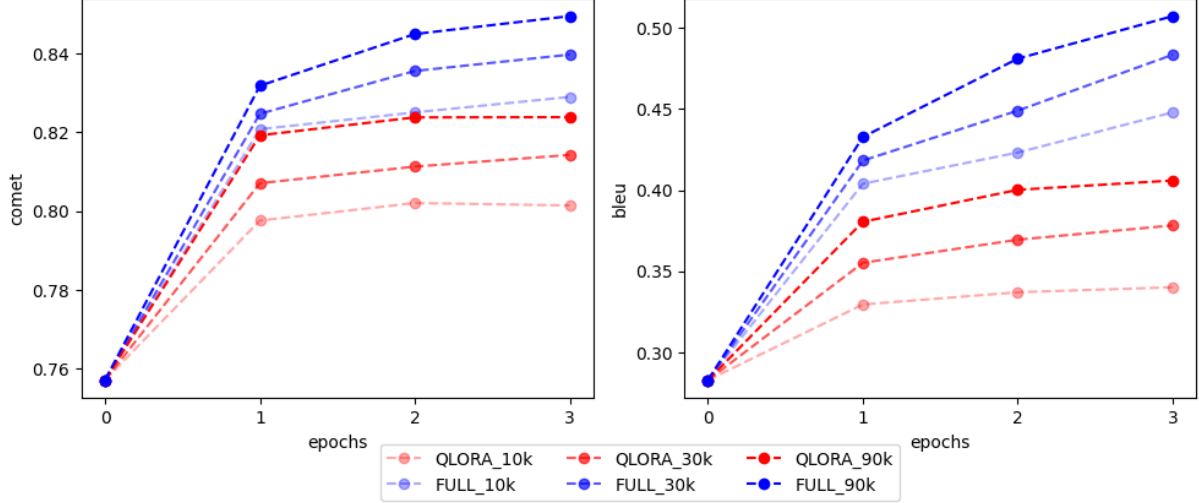
Figure 4: Effect of increasing training data and number of epochs on BLEU and COMET scores for the SAP domain test set.

all four GPUs for a single copy of the model.

5.3 Budget Efficient Training

When confronted with a fixed training budget, the pivotal decision arises between allocating resources to acquiring more training data or investing in multiple epochs. Iterating over the same data across multiple epochs expedites model fitting but poses the potential threat of overfitting, as the model may become too closely tailored to the training set. Conversely, augmenting the volume of training data holds the promise of bolstering the model's generalization capacity, yet it introduces the risk of underfitting.

Since full fine-tuning and QLoRA differ substantially in terms of trainable parameters and therefore also in expected time to reach training convergence, we investigate the effect of data and epoch increase on model performance. Figure 4 shows that QLoRA fine-tuning does not benefit all too much from an increase of epochs. After around two epochs, the test set performance is already saturated. More importantly, however, increasing the amount of training data by a factor of three provides a substantially larger boost in performance than tripling the training epochs. Therefore, we conclude that when fine-tuning with QLoRA it is sufficient to have the model observe each example only once. A reason to this is likely that the model cannot fit individual examples arbitrarily well as both the base model precision and the underparameterised adapters regulate model fit. The only way to increase the performance of QLoRA tuned models is therefore to increase the amount of training data

to allow the model to capture the underlying data distribution more wholly.

With full fine-tuning, on the other hand, we can observe a clear benefit when tuning the model over multiple epochs. Here, tripling the number of epochs results in equal or better performance than increasing the training data by a factor of three. With more trainable parameters, the model can improve its fit on an individual example with each visit. Naturally, increasing the epochs further and further will result in diminishing returns in terms of test set performance or might even lead to overfitting. Nonetheless, if confronted with limited computational budget, one should consider reducing the training data in favour of more than one epoch of training.

We would like to emphasize how differently the LLMs learn compared to the traditional and much smaller encoder-decoder translation models. These models are trained with substantially more data over dozens of epochs, since training is much cheaper, faster and requires comparably few parameters. In contrast, we see that the LLMs are quickly and easily adjusted to a downstream task in a few epochs and with a few thousand examples. This compensates for the higher training cost per sample due to the large model size.

5.4 Domain-Specific Translation - Appetite for Parallel Data

For both full fine-tuning and QLoRA, the test set performance continuously increases logarithmically with respect to the amount of training data. This observation stands in firm contrast to Xu et al. (2023),

who notice a lack of improvement beyond 10k translation examples when fine-tuning a Llama-2 7B model. Consequently, they argue that LLMs are not hungry for parallel data and suffer from catastrophic forgetting (French, 1999; Kirkpatrick et al., 2016) when confronted with too many examples. These conflicting observations could, for one, be explained by the larger size of the Llama-2 model employed in our experiments. After all, robustness to catastrophic forgetting scales with model size (Dyer et al., 2022).

We argue, however, that the different observations could stem from the type of training and test data, rather than the models. Xu et al. (2023) tune their model on general domain translation data, for which the LLM already contains knowledge that can be leveraged. Consequently, the fine-tuning just needs to nudge the model in the right direction to utilize this intrinsic knowledge.

In our case, however, we fine-tune the model on domain-specific data, which poses two challenges for the model. For one, the model needs to 'learn' the domain to retrieve relevant pre-trained knowledge. With an increase of training data, the model can come to a better understanding of what the domain really entails. Much more, however, the model also needs to internalize very rare or even new information it encounters during training. The update in parameters required for this in turn is much larger than the small nudge required for general domain translation.

5.5 Tackling Shortcomings of (Q)LoRA

The experimental results show a substantial gap in performance between QLoRA tuning and full fine-tuning. To attempt to close this gap, we experimented with various configurations for QLoRA training.

We note three observations based on the results in table 2. First, while applying adapters to all attention and feed forward matrices provides a substantial boost, it is still not comparable to full fine-tuning. Second, making the LoRA adapter bias terms trainable does not yield any benefit.

Finally, since the QLoRA adapters are of low rank, our suspicion was that the number of parameters is simply insufficient to learn the domain-specific intricacies of the translation data. Intuitively, with the rank approaching the full rank of the matrix and applying it to all matrices, we should also observe the performance converge towards the

Adapt. Attention	Adapt. FFN	QLoRA Bias	QLoRA Rank	BLEU
✓			8	0.406
✓			32	0.408
✓			64	0.407
	✓		8	0.381
✓	✓		8	0.458
✓	✓		64	0.456
✓	✓	✓	8	0.456

Table 2: BLEU scores on the in-domain test data for different QLoRA configurations fine-tuned with 90k parallel segments. *Adapt. Attention* signifies the low rank adaptation of the model's *query*, *key*, *value* and *out* projection matrices within the attention submodule. *Adapt. FNN* signifies the low rank adaptation of the *up*, *gate* and *down* matrices within the model's MLP submodule. *QLoRA Bias* indicates whether the low-rank adapter contains tunable bias terms. *QLoRA Rank* specifies the rank of the low-rank approximation matrices.
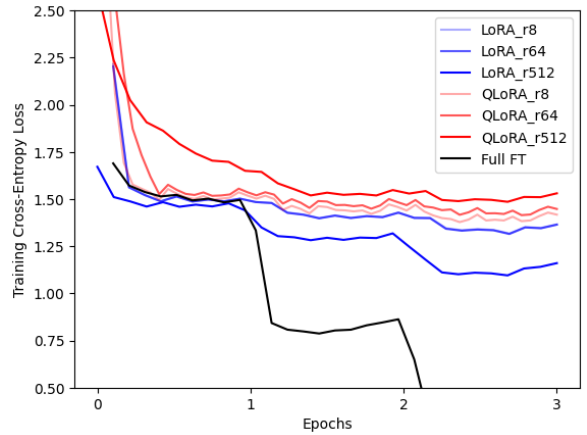

Figure 5: Effect of quantization and adapter rank on model fit.

one of the full fine-tuning. However, increasing the rank does not result in any improvement. Even worse, training runs with even higher ranks resulted in continuously degrading performance. A small grid search over learning rates and LoRA α terms could not alleviate this issue.

As figure 5 shows, QLoRA training runs are underfitting, converging quickly to a loss of about 1.5 and only very slowly beyond, regardless of adapter rank. Full fine-tuning, on the other hand, is able to fit the data much better, which becomes apparent in the dips the loss curve takes with each epoch. After all, the more often an example has been visited, the smaller the loss on it in future iterations. While the full fine-tuning model's loss curve suggests overfitting, this is not the case yet after three epochs and validation scores are substantially better than the LoRA and QLoRA runs.

Since increasing the rank of QLoRA adapters does not result in similar fitting behaviour, we cannot hold the number of parameters alone responsible for the bad model fit. Consequently, we

investigated quantization as possible culprit. For non-quantized LoRA, figure 5 shows indeed a positive correlation between adapter rank and model fit, confirming this suspicion. With sufficiently large rank, we can once again observe the desired loss-dip at epoch boundaries, indicating that the model can fit individual examples rather than just the translation task in general. This is verified by the COMET/BLEU scores starting to converge towards the full fine-tuning scores.

A hypothesis to why quantization acts as a bottleneck might be that it causes the model to lose fine-grained information. While this loss might not be apparent when prompting the model, it could become noticeable during fine-tuning. After all, fine-tuning the model allows us to better leverage the relevant pre-trained knowledge that could not be accessed as easily through prompting. If this knowledge in turn is encoded in higher precision variations in the parameters, quantization would inevitably result in its loss.

While keeping the base model unquantized improves adapter-based fine-tuning, the BLEU scores still lack behind full fine-tuning. In order to approach full fine-tuning performance, one has to increase the rank beyond 512, which negates the advantages that *low* rank adaptation would offer in the first place. The computational load for such high ranks is comparable to a full fine-tuning, since the hidden layer size for Llama-2 13B is 5120.

5.6 Pushing The Limits

Up to now we have investigated parameter efficient techniques and compared them to a full fine-tuning. The results indicate that only full fine-tuning of LLMs can possibly lead to results comparable to GPT-4 and the encoder-decoder MT system used at SAP. Therefore, we conducted a full fine-tuning with larger datasets, 200k and 400k, to push the limits. Due to a lack of improvement in general domain translation (see figure 3), we upsampled non-UI texts to diminish the dominance of simple and short UI strings. These non-UI texts are closer to general domain translation, featuring syntactically complete sentences rather than just phrases. With this change in the data mixture, we hope to see further fine-tuning improvements on both test sets.

Figure 6 shows that the domain-specific performance of the Llama-2 model approaches the one of SAP's MT with increasing amount of training
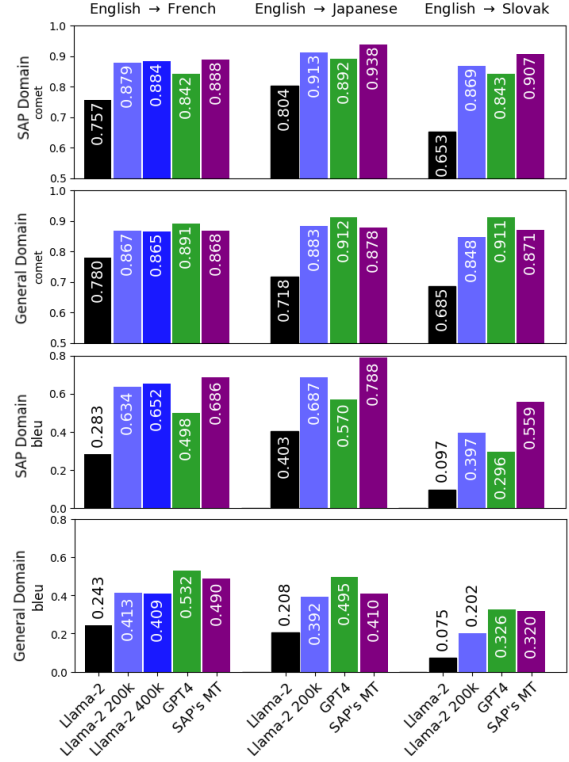

Figure 6: Llama-2 performance for larger training data sets and additional language pairs.

data. Further increasing the amount of training data will likely allow the fine-tuned LLM to surpass the MT system on the domain-specific test set. However, the same cannot be stated for general domain performance. While increasing the proportion of non-UI training data helped exceed the general domain performance observed in figure 3, we observe that further doubling the total training data does not lead to further improvement.

Possibly, further data balancing and increases in training data could allow us to fine-tune Llama-2 to match SAP's MT on both test sets. Nonetheless, it becomes apparent that downstreaming an open-source LLM to outperform a smaller dedicated translation model is **no trivial task**.

5.7 Additional Language Pairs

To complete the picture, we also conducted experiments with two additional language pairs: English → Japanese, known for its complexity, and English → Slovak, a low-resource language pair. The results are also presented in figure 6. They show the same trends as for English → French. For the general domain, the performance saturated quickly in the same regime as the SAP MT system. For Slovak and Japanese, GPT-4 performs best on the general
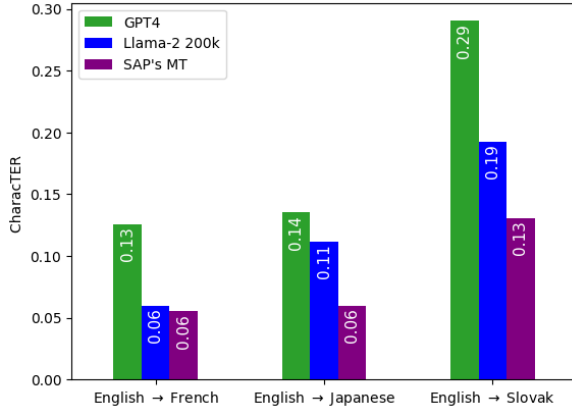
Figure 7: Human Evaluation.

domain, but as discussed before, GPT-4 has potentially seen the general domain data in training. For the SAP domain, the Llama-2 model surpasses GPT-4 and approaches the performance of the SAP's MT system with sufficient training data.

5.8 Human Evaluation

To validate the automatic scores, we conducted a human evaluation on 300 sentences randomly selected from the SAP test set for all three language pairs. The translations generated by GPT-4, Llama-2 200K, and SAP's MT were post-edited by two professional translators familiar with the SAP domain. We used CharacTER (Wang et al., 2016) to calculate the edit distance between the MT output and the post-edited version, and averaged the results from both translators. A lower edit distance suggests a higher quality of translation. The results, as shown in figure 7, largely corroborate the automatic metrics reported in figure 6.

6 Conclusion

We have shown that Llama-2 13B shows great potential for domain-specific translation fine-tuning and can substantially improve over its zero-shot performance. However, doing so is no trivial task. We find that few-shot prompting is not sufficient to close the performance gap to productive systems in the SAP domain. Parameter efficient fine-tuning with low rank adaptation fails to internalize domain-specific phenomena and therefore cannot compete with a full fine-tuning. Full fine-tuning, however, requires substantially more GPU compute power, which in turn is reflected in increased monetary costs.

While we were able to approach the performance of the comparably small encoder-decoder MT sys-

tem trained and employed at SAP by continuously increasing training data, we were unable to surpass it. Considering the much higher monetary inference costs and lower inference speed of the LLM compared to the MT model, the benefit of switching systems is not immediately obvious, especially when separate models would be hosted for various languages.

It is without doubt, however, that with rapidly improving released open-source models the performance for domain-specific LLM translation fine-tuning is bound to increase as well. Therefore, a continuous investigation into the translation capabilities of future open-source models is imperative.

7 Limitations and Future Work

Since LoRA is not sufficient to close the gap to full fine-tuning, we believe that multilingual fine-tuning could be a way to achieve better performance. This approach would also be more cost-efficient than fine-tuning LLMs individually for each language pair, considering their large parameter size. The emergence of multilingual models like Üstün et al. (2024) or Alves et al. (2024) makes this route even more promising. Particularly, the Tower model, which is based on Llama-2, seems to be a promising candidate. We plan to conduct multilingual fine-tuning experiments with this model in the future.

Finally, it should be pointed out that in our study, we only fine-tuned and evaluated the models at the sentence level. However, with their large context windows, LLMs are not limited to sentence-level translations and could translate whole documents. This could be especially beneficial in the SAP domain, where the consistent translation of whole technical documents is important. Progress might be more promising with fine-tuning and inference at the document level.

References

Alves, Duarte, Nuno Guerreiro, João Alves, José Pombal, Ricardo Rei, José de Souza, Pierre Colombo, and Andre Martins. 2023. Steering large language models for machine translation with finetuning and in-context learning. In Bouamor, Houda, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11127–11148, Singapore, December. Association for Computational Linguistics.

Alves, Duarte M., José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal,

Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks.

Babakniya, Sara, Ahmed Roushdy Elkordy, Yahya H. Ezzeldin, Qingfeng Liu, Kee-Bong Song, Mostafa El-Khamy, and Salman Avestimehr. 2023. Slora: Federated parameter efficient fine-tuning of language models.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Larochelle, H., M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Buschbeck, Bianka, Jennifer Mell, Miriam Exel, and Matthias Huck. 2022. "Hi, how can I help you?" Improving machine translation of conversational content in a business context. In Moniz, Helena, Lieve Macken, Andrew Rufener, Loïc Barrault, Marta R. Costa-jussà, Christophe Declercq, Maarit Koponen, Ellie Kemp, Spyridon Pilos, Mikel L. Forcada, Carolina Scarton, Joachim Van den Bogaert, Joke Daems, Arda Tezcan, Bram Vanroy, and Margot Fonteyne, editors, *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 191–200, Ghent, Belgium, June. European Association for Machine Translation.

Chen, Guanzheng, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In Goldberg, Yoav, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626, Abu Dhabi, United Arab Emirates, December. Association for Computational Linguistics.

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs.

Dyer, Ethan, Aitor Lewkowycz, and Vinay Ramasesh. 2022. Effect of scale on catastrophic forgetting in neural networks. In *ICLR*.

French, Robert M. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135.

Goyal, Naman, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Hendy, Amr, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are GPT models at machine translation? A comprehensive evaluation.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Kirkpatrick, James, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Laurençon, Hugo, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gérard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Romero Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Vu Minh Chien, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Ifeoluwa Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Luccioni, and Yacine Jernite. 2022. The bigscience ROOTS corpus: A 1.6TB composite multilingual dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Li, Jiahuan, Hao Zhou, Shujian Huang, Shanbo Cheng, and Jiajun Chen. 2023. Eliciting the translation ability of large language models via multilingual finetuning with translation instructions.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang,

Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 technical report.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Isabelle, Pierre, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Penedo, Guilherme, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Rei, Ricardo, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In Koehn, Philipp, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid), December. Association for Computational Linguistics.

Reimers, Nils and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Søgaard, Anders, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In Merlo, Paola, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online, April. Association for Computational Linguistics.

Sun, Xianghui, Yunjie Ji, Baochang Ma, and Xiangang Li. 2023. A comparative study between full-parameter and lora-based fine-tuning on chinese instruction data for instruction following large language model.

Wang, Weiyue, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In Bojar, Ondřej, Christian Buck, Rajen Chatterjee, Christian Federmann, Liane Guillou, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Pavel Pecina, Martin Popel, Philipp Koehn, Christof Monz, Matteo Negri, Matt Post, Lucia Specia, Karin Verspoor, Jörg Tiedemann, and Marco Turchi, editors, *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany, August. Association for Computational Linguistics.

Xu, Haoran, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models.

Zhang, Xuan, Navid Rajabi, Kevin Duh, and Philipp Koehn. 2023. Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with QLoRA. In Koehn, Philipp, Barry Haddow, Tom Kocmi, and Christof Monz, editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481, Singapore, December. Association for Computational Linguistics.

Zhu, Wenhao, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis.

Üstün, Ahmet, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model.