
How Much Data is Enough Data? Fine-Tuning Large Language Models for In-House Translation: Performance Evaluation Across Multiple Dataset Sizes

Inacio Vieira*

inacio@gmail.com

Department of Computing, Dublin City University, Dublin, Ireland & Alpha CRC, Cambridge, UK

Will Allred*

william.allred2@mail.dcu.ie

Department of Computing, Dublin City University, Dublin, Ireland

Séamus Lankford

seamus.lankford@adaptcentre.ie

Department of Computer Science, Munster Technological University, Cork, Ireland

Sheila Castilho

sheila.castilho@adaptcentre.ie

SALIS/ADAPT Centre, Dublin City University, Dublin, Ireland

Andy Way

andy.way@adaptcentre.ie

ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland

Abstract

Decoder-only LLMs have shown impressive performance in MT due to their ability to learn from extensive datasets and generate high-quality translations. However, LLMs often struggle with the nuances and style required for organisation-specific translation. In this study, we explore the effectiveness of fine-tuning Large Language Models (LLMs), particularly Llama 3 8B Instruct, leveraging translation memories (TMs), as a valuable resource to enhance accuracy and efficiency.

We investigate the impact of fine-tuning the Llama 3 model using TMs from an organisation in the software sector. Our experiments cover five translation directions across languages of varying resource levels (English to Brazilian Portuguese, Czech, German, Finnish, and Korean). We analyse diverse sizes of training datasets (1k to 207k segments) to evaluate their influence on translation quality. We fine-tune separate models for each training set and evaluate their performance based on automatic metrics, BLEU, chrF++, TER, and COMET.

Our findings reveal improvement in translation performance with larger datasets across all metrics. On average, BLEU and COMET scores increase by 13 and 25 points, respectively, on the largest training set against the baseline model. Notably, there is a performance deterioration in comparison with the baseline model when fine-tuning on only 1k and 2k examples; however, we observe a substantial improvement as the training dataset size increases. The study highlights the potential of integrating TMs with LLMs to create bespoke translation models tailored to the specific needs of businesses, thus enhancing translation quality and reducing turn-around times. This approach offers a valuable insight for organisations seeking to leverage TMs and LLMs for optimal translation outcomes, especially in narrower domains.

1 Introduction

In recent years, decoder-only large language models (LLMs) have revolutionised the machine transla-

tion (MT) field due to their ability to learn from vast amounts of data and generate high-quality translations (Alves et al., 2023a; Moslem et al., 2023a; Mu et al., 2023; Robinson et al., 2023; Zhu et al.,

Datasets	Segments
Aligned Training	1000, 2000, 5000, 10000, 14688
Dev	1837
Test	1353

Table 1: Segment counts for the various aligned training dataset sizes, the development set, and the test set.

2023; Lyu et al., 2024). LLMs, such as Llama 3 8B Instruct,¹ have shown impressive capabilities in adapting to translation tasks, generating human-like accurate output, making them invaluable tools for the sector (Li et al., 2023; Moslem et al., 2023b; Lyu et al., 2024). However, out-of-the-box LLMs do not always capture all the nuances, appropriate tone, and terminology required for specialised or organisation-specific translations (Moslem et al., 2022; Alves et al., 2023b; Zheng et al., 2024). This is where translation memories (TMs) offer a potential solution.

A TM is a database that stores previously human-translated segments and their respective translations. They are particularly useful to language service providers (LSPs) as they deal with repetitive content and organisation-specific style and terminology, enhancing the efficiency and accuracy of translations (Bloodgood and Strauss, 2014; Bulte and Tezcan, 2019; Moslem et al., 2023a). Therefore, the integration of TMs and LLMs can create models that better understand organisational requirements and lead to higher quality outputs and reduced turnaround times. However, this approach depends on several factors, like the amount, quality and specificity of the TMs used as training data for fine-tuning.

Previous work explored fine-tuning of models with TM for translation for specific domains and the benefit that offers to performance (Haque et al., 2020; Moslem et al., 2022). Accordingly, TM provides much value because of its high quality and domain relevance (Bulte and Tezcan, 2019; Xu et al., 2020; Cai et al., 2021; Knowles and Littell, 2022). This research highlights the gains available when leveraging existing TMs during the fine-tuning process of LLMs.

In this paper, we investigate a real-life scenario where we fine-tune Llama 3 8B Instruct (Llama

Team, 2024) using TMs from a specific organisation. Additionally, since increasing the fine-tuning data requires dedicating more resources and time, we explore different dataset sizes to evaluate their impact on translation quality and identify the most efficient return on investment. We conduct experiments in five translation directions (from English) on languages of varying resource level (Brazilian Portuguese (PT-BR), Czech (CS), German (DE), Finnish (FI), and Korean (KO)). This approach can lead to bespoke translation models that cater to the unique needs of different companies when compared to generic LLMs.

2 Methodology

2.1 Data

The raw dataset consists of TMs from an anonymous organisation that operates in the software sector. The three datasets employed cover knowledge base, mobile user interface, and mobile reference materials.

The five target languages dataset (PT-BR, CS, DE, FI, and KO) are filtered to remove duplicates, source-copies, and segments over 150 words to ensure none would go over the maximum length set during training. All HTML tags are removed, and double spaces are converted to single spaces. Any rows containing only dates, version numbers, or any programming language are also removed. Rows are then randomly shuffled to mitigate any temporal bias that could arise from the chronological order of the data, ensure the model does not memorise sequences, and prevent the evaluation set from being biased towards a particular section of the data.

The dataset is then transformed into an interlingual aligned dataset for all five target languages where any rows with missing translations for any target languages, are dropped. This results in a

*These authors contributed equally to this work

¹<https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>

Lang	Full Training Data	Dev Data	Total Segments
Brazilian Portuguese (PT-BR)	217,555	54,389	271,944
Czech (CS)	107,555	26,889	134,804
German (DE)	223,894	55,973	279,867
Finnish (FI)	207,218	51,805	259,023
Korean (KO)	162,360	40,590	202,950

Table 2: Segment counts for the full datasets used during training.

dataset where all source segments have translations available in all five target languages. The dataset is then split into training, development, and test sets, as shown in Table 1.

Further filtering is applied to the test set removing segments that had over 75% similarity with any segments in the training dataset to ensure robust testing and minimal memorisation. We measure similarity as a combination of the Levenshtein distance (Levenshtein, 1965) and a 5-gram-based similarity (Lopez-Gazpio et al., 2019). This reduced the size of the test split from 1837 to 1353. The test split with under 75% similarity was used for all experiments.

In the interest of using all the data available, we also compile all segments in a given language into a dataset for each target language. This includes any segment that would not fit the inter-lingual alignment criteria applied above. This will now be referred to as the ‘full dataset’. These larger training sets allow us to train beyond the 14.7k aligned segments and make use of the total volume of available segments in order to explore what impact that would have on results. The full training sets range from 107k (CS) to 223k (DE) examples, as shown in Table 2.

2.2 Model

We use the Llama 3 8B Instruct model and its associated tokenizer (Llama Team, 2024). The decision between the Instruct and the base model is based on an extensive MT evaluation of Llama 3 models (Wu et al., 2024) using the Flores-200² dataset (Guzmán et al., 2019; Costa-jussà et al., 2022). Even though Wu (2024) dealt with the opposite language direc-

tion (X to English), we consider the close results between Instruct and the base model involving the five languages included in this paper to be a good indicator of proximity in performance between the models. Our baseline consists of the test set metric results obtained from the out-of-the-box Llama 3 8B Instruct model. We use QLoRA (Hu et al., 2021; Dettmers et al., 2023) for efficient fine-tuning with 4-bit quantisation using Hugging Face Transformers. We perform fine-tuning on a high performance cluster with four A100-SXM4-80GB GPUs. From Hugging Face, we leverage the Supervised Fine-Tuning Training (SFTTrainer),³ which is a wrapper of the Trainer class⁴ optimized for fine-tuning language models like Llama. On the largest dataset size, fine-tuning takes approximately 2.3 hours (Appendix A).

2.3 Inference

2.3.1 Prompting

At inference time, we use many of the recommended parameters from previous work (Moslem et al., 2023b) and model documentation to produce translation outputs from the baseline model and the fine-tuned versions (cf. Appendix C). Meta’s Llama 3 documentation⁵ provides a recommended prompt format and instructions to implement special tokens during inference and training (Llama Team, 2024).

The prompt and the source segment were passed to the model for inference to obtain each translation. This constitutes zero-shot as it did not include examples in the prompt (Zhang et al., 2023). A JSON scheme (`{“translation”: “string”}`) was also added to the prompt in order to obtain a structured output (Wu et al., 2024). During training, the

²github.com/facebookresearch/flores/blob/main/flores200/README.md

³https://huggingface.co/docs/trl/en/sft_trainer

⁴https://huggingface.co/docs/transformers/en/main_classes/trainer

⁵<https://llama.meta.com/docs/model-cards-and-prompt-formats/meta-llama-3/>

BitsAndBytes Quantisation Configuration	
load_in_4bit	True
bnb_4bit_quant_type	“nf4”
bnb_4bit_use_double_quant	True
bnb_4bit_compute_dtype	torch.bfloat16
PEFT LoRA Configuration	
low-rank matrix dimension (r)	64
scaling factor (lora_alpha)	16
dropout probability (lora_dropout)	0.1
training of bias parameters (bias)	“none”
Training Arguments	
batch size for training and evaluation	32 examples
learning rate	2e-3
lr_scheduler_type	“constant”
bf16	True

Table 3: Fine-tuning hyperparameters.

same format was applied with the addition of the specific EOS token (`<|end_of_text|>`) as recommended by Meta’s documentation (cf. Appendix B).

2.3.2 Translation

In order to obtain higher efficiency, both baseline and fine-tuned models are converted to the CTranslate2⁶ (Klein et al., 2020) format (with 8-bit quantisation) and provided with parameters for inference (cf. Appendix C).

2.3.3 Stopping Criteria and Post-processing

In early experiments, we observe frequent instances of overgeneration; an issue recently explored further by Zheng et al. (2024). By using “}assistant” as a *stop_token* in our *stopping_criteria*, we find much less post-processing is required in order to obtain the pure translation.

Our post-processing consists of extracting the translation by removing the ‘{“translation”: “’ prefix and the trailing ‘”}’. The newline characters are replaced by spaces. On some occasions, especially in the models produced by the smaller training datasets (1k and 2k examples), further cleaning is required as the model inadvertently overgenerated some HTML tags like ‘
’ and ‘<p>’. This is

important to properly assess the translation quality.

2.4 Evaluation

To evaluate the performance of our models, we report BLEU (Papineni et al., 2002), chrF++ (Popović, 2017), TER (Snover et al., 2006) via sacreBLEU,⁷ and COMET⁸ (Rei et al., 2020). We use multiple metrics to make our experiments more comparable to a wider variety of work and to provide insight into certain aspects of performance.

It is important to note that the experiment aims to show the training efficiency of the PEFT fine-tuning method and its ability to approximate the model’s translating capabilities to the training material. Therefore, we pay special attention to the automatic metrics measuring *n*-gram differences and edits (BLEU, chrF++, TER) whilst still considering the quality estimation aspect of COMET as a means of comparing inter-source languages and other similar research. Our results are compared to those obtained from the baseline model, an out-of-the-box Llama 3 8B Instruct model, and to GPT 3.5. We also ask five professional translators to post-edit 100 translations from the best-performing model into their language pair. They also answer a questionnaire about

⁶<https://github.com/OpenNMT/CTranslate2>

⁷<https://github.com/mjpost/sacrebleu>

⁸wmt20-comet-da, <https://github.com/Unbabel/COMET>

the quality of the automatically translated segments. The questionnaire asks for comments on the quality of the translations.

3 Results and Discussion

The results in Table 4 show an increase in performance across all the languages for all datasets with more than 5k segments compared to the baseline. The fully aligned 14.7k dataset sees a BLEU score increase of 4.8 points or relative increase of 17.42% on average over the baseline, over all target languages, while chrF++ and COMET increases 7.1 and 16.9, respectively. Similarly, TER decreases by 9 points. The 100k+ datasets also demonstrate consistent performance gains with an average increase of 13.7 BLEU, 12.7 chrF++, and 25 COMET, while TER decreases to 15.5.

To provide a point of comparison, we evaluate the performance of GPT-3.5⁹ on our test set. While GPT-3.5 outperforms our highest-performing model in BLEU and chrF++ for DE and FI, the 100k+ datasets often surpass GPT-3.5 in other languages and metrics. This demonstrates the effectiveness of creating bespoke models through fine-tuning mid-sized LLMs when leveraging domain-specific data. Targeted fine-tuning can yield competitive or superior results compared to larger, general-purpose models like GPT-3.5.

3.1 Small Dataset Deterioration

Regarding translation quality across different training data sizes, we note a deterioration in quality for models trained on the smaller datasets (1k and 2k) in relation to the baseline. Despite a smooth reduction in both training and evaluation loss during training across all sizes, these smaller datasets still lead to poorer performance on all metrics. This can be due to the fact that the 1k and 2k datasets are insufficient to offer the models a wide enough variety of examples, leading to overfitting where the model performs well on training but poorly on the unseen test data (Barone et al., 2017; Atrio and Popescu-Belis, 2022; Garcia et al., 2023; Ramírez Atrio, 2023).

It is possible that the lack of diversity in the smaller models fails to capture the range of linguistic and translation nuances present in the test data which hinders the model’s ability to generalise beyond the specific examples seen during train-

ing. Furthermore, the smaller datasets may make the models more susceptible to noise, such as translation errors or inconsistencies, leading to the learning of incorrect patterns and degrading performance on the test data, affecting the automatic metrics results, while the loss continues to drop due to fitting noisy data (Barone et al., 2017; Atrio and Popescu-Belis, 2022; Ramírez Atrio, 2023).

Another possible explanation for the deterioration is a decrease in training data quality in the 1k and 2k dataset sizes. To examine this, we use COMET-Kiwi (Rei et al., 2023), a popular quality estimation metric, to evaluate the quality of the training data. The scores are consistent for each language with variations within a narrow range of 1-2 points (cf. Appendix D). For example, FI has the highest variation with a maximum score of 79.58 (1k and 14.7K) and a minimum score of 78.12 (5k), resulting in a range of only 1.46 points. The minimal variation in score indicates consistent data quality across all dataset sizes for each language. Therefore, the deterioration in performance is unlikely to be due to a decrease in data quality for the 1k and 2k training data sizes.

Hyperparameter fine-tuning could be employed to mitigate this early deterioration in situations where only small datasets are available. This may include dropout or other regularisation techniques to prevent overfitting on small training sets. Adjustment of the learning rate, batch sizes and QLoRA hyperparameters should also be explored to deal with this specific case of deterioration (Barone et al., 2017; Atrio and Popescu-Belis, 2022; Dettmers et al., 2023; Ramírez Atrio, 2023).

Overall, a different approach is required in order to obtain gains when the training data is scarce. Our experiments suggest the need for at least 5k examples to achieve an improvement in metrics under the hyper-specific domain and circumstances we explore.

The issues above seem to be mitigated on the larger sets whilst maintaining the same hyperparameters as previously reported (cf. Table 3). We observe performance recovery on 5k examples, overtaking the baseline model, then consistently improving over all metrics as dataset size increases, and achieving increasingly impressive results across all metrics when training on anything above the 10k

⁹<https://chat.openai.com/>

Lang	Data Size	BLEU \uparrow	chrF++ \uparrow	TER \downarrow	COMET \uparrow
PT-BR	GPT 3.5	56.50	76.33	32.03	86.02
	Baseline	48.25	69.21	39.36	77.28
	1k	48.00	69.34	40.11	78.28
	2k	<u>46.04</u>	<u>67.93</u>	<u>44.09</u>	<u>75.70</u>
	5k	49.73	69.92	38.03	80.80
	10k	50.90	70.92	35.96	86.15
	14.7k	53.42	73.07	32.92	89.18
	100k+	62.45	78.57	26.20	95.98
CS	GPT 3.5	31.78	55.02	58.17	72.99
	Baseline	26.25	49.97	63.27	62.43
	1k	26.53	50.15	64.97	64.20
	2k	<u>25.23</u>	<u>48.35</u>	<u>68.76</u>	<u>58.28</u>
	5k	27.57	51.35	62.84	66.85
	10k	27.96	52.40	63.26	66.62
	14.7k	31.57	54.75	60.07	73.73
	100k+	39.72	61.45	52.00	84.22
DE	GPT 3.5	42.41	65.88	50.07	65.31
	Baseline	34.32	59.16	57.60	58.36
	1k	34.58	59.07	58.42	60.86
	2k	<u>32.45</u>	<u>57.08</u>	<u>62.93</u>	<u>53.87</u>
	5k	35.31	59.37	56.19	63.66
	10k	37.23	60.58	53.59	66.82
	14.7k	37.88	61.08	52.71	68.50
	100k+	42.27	65.15	48.59	73.01
FI	GPT 3.5	33.80	59.18	58.29	83.84
	Baseline	23.97	49.70	70.36	62.64
	1k	24.14	49.48	71.18	65.22
	2k	<u>19.07</u>	<u>46.97</u>	<u>83.00</u>	<u>58.34</u>
	5k	22.05	47.32	75.12	60.54
	10k	25.88	50.71	65.99	74.91
	14.7k	26.48	51.32	64.91	73.66
	100k+	31.71	57.13	59.72	84.71
KO	GPT 3.5	33.07	49.72	60.60	63.28
	Baseline	20.81	<u>35.37</u>	77.95	36.45
	1k	20.12	42.16	<u>83.37</u>	35.24
	2k	<u>19.25</u>	41.13	82.48	<u>26.03</u>
	5k	28.60	46.84	65.42	54.17
	10k	31.36	52.62	60.86	70.56
	14.7k	28.15	58.88	53.11	76.65
	100k+	45.80	64.81	44.73	84.30

Table 4: Evaluation results of fine-tuning Llama 3 8B on datasets of various sizes. **Bold** text indicates the best score. The models trained on the largest dataset (100k+) perform the best. The scores deteriorate from the baseline for 1k and 2k but recover and increase from 5k onward. Underlined text indicates the worst scores.

sets and excelling on the 100k+ sets.

3.2 Resource Level

It is interesting to note that the performance for KO has improved after the 14.7k fine-tuning and becomes comparable to or better than the performance of the other language directions, despite the lower initial baseline score across all metrics. For instance, the COMET score for the KO baseline is 36.5 while the average for all other languages is 57.7. We find that the lower resource languages (KO being the lowest of the target languages explored) have the highest relative gains, turning around a very poor baseline across all metrics. The COMET score for KO increased to 84.3 compared to the average of 84.5 in the 100k+ datasets for PR-BR, DE, FI, and CS, resulting in KO’s comparable performance to the high resource languages, i.e. PT-BR and DE.

These results probably relate not only to the resource level of the language but also to the amount of Korean data in the Llama 3 training recipe. According to MetaAI, “over 5% of the Llama 3 pre-training dataset consists of high-quality non-English data that covers over 30 languages” (Meta, 2024). While the Llama Team provides more detail on the training and data mix Llama 3, the exact proportion of Korean data is not discussed (Llama Team, 2024). Our baseline metrics suggest that Korean does not feature highly on that list given that it scores significantly lower than all other languages. This might be attributed to the fact that there were not enough examples to produce a firm understanding of the language but enough to provide a foundation that heavily benefited from fine-tuning. As mentioned, this is an assumption as we lack sufficiently detailed information on the training recipe.

When looking at the target languages, we note that PT-BR shows the best performance at 14.7k and 100k+ dataset. This indicates that, even for a well-resourced language, the foundation model gained a strong understanding of the language during pre-training. However, it did not seem to benefit as much from fine-tuning as KO, a lower resource language. This corroborates the finding that resource level is a strong determinant of LLM MT performance (Robinson et al., 2023).

3.3 Human Evaluation

Regarding the human evaluation, the qualitative comments from the translators reveal that the largest model struggles with ambiguity. Evaluators men-

tion that segments that lacked complete information needed to be completely reworked. For example, the segment, “Get basic, step-by-step instructions to learn” lacks a final object, which impacts the translation. While human translators often face and resolve such ambiguities through research or decision-making with incomplete information, the model processes segments in isolation, unable to access potentially clarifying context from adjacent segments. This limitation provides insight into the model’s performance in real-world translation scenarios.

4 Conclusions

Fine-tuning on TMs has been demonstrated to enhance the performance of LLMs in MT tasks. In this paper, we investigate the relationship between automatic metric results and training set sizes to identify the optimal balance where resource investment yields the most significant improvements in translation quality. In our experiments, it has become evident that fine-tuning on training datasets whose size is larger than 5k examples returned increasingly better results in 19 out of the 20 language-training set size combinations explored.

By leveraging TMs, the model becomes more adept at recognising and reproducing previously translated segments, their style, and terminology. Furthermore, fine-tuning on TM data helps the model adapt to specialised fields.

The test and training sets used come from a much narrower corpus of data than in similar experiments that deal with wider domains, i.e. medicine (Moslem et al., 2023b). The hyper-specific nature of the training data employed in our approach may partly explain the promising results. We therefore leverage the advantage that smaller models licensed for business-use offer; they can be adapted several times over for narrow and specific domains, as well as multiple languages with little investment, instead of aiming for a more general purpose or multilingual model. The hyper-specific purpose of our trained model, i.e. one language direction and a narrow domain, suits the size and easiness of training of an 8B parameter mode.

Being a commonly experienced scenario in the localization industry, this is an under-explored approach that organisations could be pursuing in order to make the most out of their access to TMs and LLMs for MT in order to obtain the best possible re-

turn on investment when leveraging their previously human-translated material.

Low-resource languages seem to be in a perfect position to benefit from leveraging small business-friendly models, like Llama 3 8B. The gains in automatic metric results for KO are substantially higher for high resource languages, like PT-BR and DE, returning the highest increase in performance compared to the metrics obtained from training on similar set sizes in those languages. KO observes an increase of 130% on COMET from the baseline to the 100k+ dataset, whereas the average increase amongst the other target languages is 46% (cf. Table 4).

It is important to mention that, just as Wu (2024) acknowledges the FLORES-200 dataset leakage into Llama 3, it is possible that some of our test set was also scraped by the Llama 3 models, as parts of the material were published online prior to the Llama 3 family’s pre-training. We face the same challenge as the whole AI researching community, forced to either constantly come up with new test sets or simply acknowledge the potential leakage of test data (Xu et al., 2024). We urge large tech companies to disclose at a minimum the test sets that were not ingested by their models for the benefit of the whole community. We acknowledge the Llama Team’s leadership in this area (Llama Team, 2024).

5 Future Work

Future work in the area may benefit from the introduction of checkpoints during training and subsequent intermediate evaluation would enable the visualisation of a clearer learning curve, and the identification of potential dips in performance and points of diminishing returns. This approach would facilitate the analysis and allow for a finer and more efficient evaluation process.

In the future, we aim to obtain a bespoke test set directly from the organisation that owns the TMs. This tailored test set would consist of examples specifically designed in-house according to strict guidelines, ensuring they are completely original and reflective of the organisation’s unique requirements and style. By using a bespoke and unseen test set, we can more accurately assess the performance of our fine-tuned models in a real-world context.

Finally, further investigation is required with

regard to the training hyperparameters across the different dataset sizes in order to obtain better results with smaller training sets under 5k examples. Several strategies can be explored to optimise performance on smaller datasets. Adjustments such as modifying the dropout rates to prevent overfitting, applying regularisation techniques to enhance model generalisation, and fine-tuning the learning rate to ensure efficient convergence can be particularly beneficial in this case.

6 Acknowledgements

This research is supported by Science Foundation Ireland through ADAPT Centre (Grant No. 13/RC/2106) (www.adaptcentre.ie) at Dublin City University. We thank Alpha-CRC for their essential collaboration.

References

- Alves, D., Guerreiro, N., Alves, J., Pombal, J., Rei, R., de Souza, J., Colombo, P., and Martins, A. (2023a). Steering large language models for machine translation with finetuning and in-context learning. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11127–11148, Singapore. Association for Computational Linguistics.
- Alves, D., Guerreiro, N., Alves, J., Pombal, J., Rei, R., de Souza, J., Colombo, P., and Martins, A. (2023b). Steering Large Language Models for Machine Translation with Finetuning and In-Context Learning. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11127–11148, Singapore. Association for Computational Linguistics.
- Atrio, À. R. and Popescu-Belis, A. (2022). On the Interaction of Regularization Factors in Low-resource Neural Machine Translation. In Moniz, H., Macken, L., Rufener, A., Barrault, L., Costa-jussà, M. R., Declercq, C., Koponen, M., Kemp, E., Pilos, S., Forcada, M. L., Scarton, C., Van den Bogaert, J., Daems, J., Tezcan, A., Vanroy, B., and Fonteyne, M., editors, *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 111–120, Ghent, Belgium. European Association for Machine Translation.
- Barone, A. V. M., Haddow, B., Germann, U.,

- and Sennrich, R. (2017). Regularization techniques for fine-tuning in neural machine translation. arXiv:1707.09920 [cs].
- Bloodgood, M. and Strauss, B. (2014). Translation memory retrieval methods. In Wintner, S., Goldwater, S., and Riezler, S., editors, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 202–210, Gothenburg, Sweden. Association for Computational Linguistics.
- Bulte, B. and Tezcan, A. (2019). Neural Fuzzy Repair: Integrating Fuzzy Matches into Neural Machine Translation. In Korhonen, A., Traum, D., and Márquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.
- Cai, D., Wang, Y., Li, H., Lam, W., and Liu, L. (2021). Neural Machine Translation with Monolingual Translation Memory. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7307–7318, Online. Association for Computational Linguistics.
- Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A., Akula, B., Barrault, L., Gonzalez, G. M., Hansanti, P., Hoffman, J., Jarrett, S., Sadagopan, K. R., Rowe, D., Spruit, S., Tran, C., Andrews, P., Ayan, N. F., Bhosale, S., Edunov, S., Fan, A., Gao, C., Goswami, V., Guzmán, F., Koehn, P., Mourachko, A., Ropers, C., Saleem, S., Schwenk, H., and Wang, J. (2022). No Language Left Behind: Scaling human-centered machine translation.
- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314 [cs].
- Garcia, X., Bansal, Y., Cherry, C., Foster, G., Krikun, M., Johnson, M., and Firat, O. (2023). The Unreasonable Effectiveness of Few-shot Learning for Machine Translation. In *Proceedings of the 40th International Conference on Machine Learning*, pages 10867–10878. PMLR. ISSN: 2640-3498.
- Guzmán, F., Chen, P.-J., Ott, M., Pino, J., Lample, G., Koehn, P., Chaudhary, V., and Ranzato, M. (2019). The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.
- Haque, R., Moslem, Y., and Way, A. (2020). Terminology-Aware Sentence Mining for NMT Domain Adaptation: ADAPT’s Submission to the AdapMT 2020 English-to-Hindi AI Translation Shared Task. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON): Adap-MT 2020 Shared Task*, pages 17–23, Patna, India. NLP Association of India (NLP AI).
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs].
- Klein, G., Zhang, D., Chouteau, C., Crego, J., and Senellart, J. (2020). Efficient and high-quality neural machine translation with OpenNMT. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 211–217, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Knowles, R. and Littell, P. (2022). Translation Memories as Baselines for Low-Resource Machine Translation. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Odijk, J., and Piperidis, S., editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6759–6767, Marseille, France. European Language Resources Association.
- Levenshtein, V. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*.
- Li, J., Zhou, H., Huang, S., Cheng, S., and Chen, J. (2023). Eliciting the Translation Ability of Large Language Models via Multilingual Finetuning with Translation Instructions. arXiv:2305.15083 [cs].
- Llama Team (2024). The Llama 3 Herd of Models.

- Lopez-Gazpio, I., Maritxalar, M., Lapata, M., and Agirre, E. (2019). Word n-gram attention models for sentence similarity and inference. *Expert Systems with Applications*, 132:1–11.
- Lyu, C., Du, Z., Xu, J., Duan, Y., Wu, M., Lynn, T., Aji, A. F., Wong, D. F., and Wang, L. (2024). A Paradigm Shift: The Future of Machine Translation Lies with Large Language Models. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1339–1352, Torino, Italia. ELRA and ICCL.
- Meta (2024). Introducing meta llama 3: The most capable openly available llm to date. <https://ai.meta.com/blog/meta-llama-3/>.
- Moslem, Y., Haque, R., Kelleher, J., and Way, A. (2022). Domain-Specific Text Generation for Machine Translation. In Duh, K. and Guzmán, F., editors, *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 14–30, Orlando, USA. Association for Machine Translation in the Americas.
- Moslem, Y., Haque, R., Kelleher, J. D., and Way, A. (2023a). Adaptive Machine Translation with Large Language Models. In Nurminen, M., Brenner, J., Koponen, M., Latomaa, S., Mikhailov, M., Schierl, F., Ranasinghe, T., Vanmassenhove, E., Vidal, S. A., Aranberri, N., Nunziatini, M., Escartín, C. P., Forcada, M., Popovic, M., Scarton, C., and Moniz, H., editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 227–237, Tampere, Finland. European Association for Machine Translation.
- Moslem, Y., Haque, R., and Way, A. (2023b). Fine-tuning Large Language Models for Adaptive Machine Translation. arXiv:2312.12740 [cs].
- Mu, Y., Reheman, A., Cao, Z., Fan, Y., Li, B., Li, Y., Xiao, T., Zhang, C., and Zhu, J. (2023). Augmenting large language model translators via translation memories. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10287–10299, Toronto, Canada. Association for Computational Linguistics.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Popović, M. (2017). chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- Ramírez Atrio, A. (2023). *Regularization Techniques for Low-Resource Machine Translation*. PhD thesis, EPFL, Lausanne.
- Rei, R., Guerreiro, N. M., Pombal, J., van Stigt, D., Treviso, M., Coheur, L., C. de Souza, J. G., and Martins, A. (2023). Scaling up CometKiwI: Unbabel-IST 2023 Submission for the Quality Estimation Shared Task. In Koehn, P., Haddow, B., Kocmi, T., and Monz, C., editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.
- Rei, R., Stewart, C., Farinha, A. C., and Lavie, A. (2020). COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.
- Robinson, N., Ogayo, P., Mortensen, D. R., and Neubig, G. (2023). ChatGPT MT: Competitive for high- (but not low-) resource languages. In Koehn, P., Haddow, B., Kocmi, T., and Monz, C., editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 392–418, Singapore. Association for Computational Linguistics.
- Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.
- Wu, M. (2024). Evaluating the Machine Translation Capability of Llama-3 Models. Publication Title: GitHub repository.

- Wu, M., Vu, T.-T., Qu, L., Foster, G., and Haffari, G. (2024). Adapting large language models for document-level machine translation.
- Xu, J., Crego, J., and Senellart, J. (2020). Boosting Neural Machine Translation with Similar Translations. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.
- Xu, R., Wang, Z., Fan, R.-Z., and Liu, P. (2024). Benchmarking benchmark leakage in large language models.
- Zhang, X., Rajabi, N., Duh, K., and Koehn, P. (2023). Machine Translation with Large Language Models: Prompting, Few-shot Learning, and Fine-tuning with QLoRA. In Koehn, P., Haddow, B., Kocmi, T., and Monz, C., editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481, Singapore. Association for Computational Linguistics.
- Zheng, J., Hong, H., Wang, X., Su, J., Liang, Y., and Wu, S. (2024). Fine-tuning Large Language Models for Domain-specific Machine Translation. arXiv:2402.15061 [cs].
- Zhu, W., Liu, H., Dong, Q., Xu, J., Huang, S., Kong, L., Chen, J., and Li, L. (2023). Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis. arXiv:2304.04675 [cs].

A Appendix A

Lang	Size	Loss	Dev Loss	Train Runtime
PT-BR	1k	1.4922	1.0706	69
	2k	0.8271	0.8290	120
	5k	0.7506	0.6325	290
	10k	0.4222	0.5337	551
	14.7k	0.4918	0.4714	820
	100k+	0.6031	0.5964	8423
CS	1k	1.5428	1.0795	70
	2k	0.9446	0.8880	124
	5k	0.6643	0.6586	296
	10k	0.6475	0.5880	574
	14.7k	0.5346	0.5169	837
	100k+	0.5600	0.4800	8000*
FI	1k	1.5342	1.1519	71
	2k	0.9631	0.9602	125
	5k	0.5876	0.6286	302
	10k	0.5662	0.5874	589
	14.7k	0.3996	0.5138	866
	100k+	0.5964	0.5867	8241
DE	1k	1.5551	1.1397	69
	2k	0.9591	0.9301	121
	5k	0.4371	0.6426	290
	10k	0.4553	0.5639	550
	14.7k	0.5310	0.5037	819
	100k+	0.6672	0.6603	8000*
KO	1k	1.5851	1.0651	67
	2k	0.7765	0.7733	120
	5k	0.6086	0.6340	270
	10k	0.4662	0.5666	543
	14.7k	0.4167	0.4923	807
	100k+	0.7822	0.7052	5791

Table 5: Training Details by Language. Train Runtime is measured in seconds. Starred numbers are estimates.

B Appendix B

B.1 Special Token Descriptions

`<|begin_of_text|>`: This is equivalent to the BOS token.

`<|eot_id|>`: This signifies the end of the message in a turn.

`<|start_header_id|>{role}<|end_header_id|>`: These tokens enclose the role for a particular message. The possible roles can be: system, user, assistant.

`<|end_of_text|>`: This is equivalent to the EOS token.

B.2 Prompt

< |begin_of_text| > < |start_header_id| >system< |end_header_id| >

You are a helpful AI assistant for translation from {source_language} to {target_language}. You MUST answer with the following JSON scheme: {"translation": "string"} < |eot_id| >

< |start_header_id| >user< |end_header_id| >

{source_sentence}< |eot_id| >< |start_header_id| >assistant< |end_header_id| >

B.3 Training Prompt

< |begin_of_text| > < |start_header_id| >system< |end_header_id| >

You are a helpful AI assistant for translation from {source_language} to {target_language}. You MUST answer with the following JSON scheme: {"translation": "string"} < |eot_id| >

< |start_header_id| >user< |end_header_id| > {source_sentence}< |eot_id| >

< |start_header_id| >assistant< |end_header_id| > {target_sentence}< |end_of_text| >

C Appendix C

Inference Parameters

sampling_topk	1
max_batch_size	8096
min_length	1
max_length	double the source length

Table 6: CTranslate2 Inference Parameters.

D Appendix D

Lang	Data Size	Comet-Kiwi \uparrow
PT-BR	1k	77.95
	2k	77.81
	5k	77.65
	10k	77.77
	14.7k	78.92
CS	1k	79.71
	2k	78.98
	5k	78.57
	10k	78.78
	14.7k	79.71
DE	1k	78.58
	2k	78.22
	5k	78.34
	10k	78.21
	14.7k	78.73
FI	1k	79.58
	2k	78.70
	5k	78.12
	10k	78.54
	14.7k	79.58
KO	1k	81.93
	2k	81.56
	5k	81.20
	10k	81.22
	14.7k	81.55

Table 7: Quality Evaluation results of training datasets of different sizes using Comet-Kiwi metric.