Enriching the Low-Resource Neural Machine Translation with Large Language Model

Sachin Giri **Takashi Ninomiya** **Isao Goto**
Graduate School of Science and Engineering, Ehime University
sachin.giri.cs@gmail.com, {ninomiya.takashi.mk, goto.isao.fn}@ehime-u.ac.jp

Abstract

Improving the performance of neural machine translation for low-resource languages is challenging due to the limited availability of parallel corpora. However, recently available Large Language Models (LLM) have demonstrated superior performance in various natural language processing tasks, including translation. In this work, we propose to incorporate an LLM into a Machine Translation (MT) model as a prior distribution to leverage its translation capabilities. The LLM acts as a teacher, instructing the student MT model about the target language. We conducted an experiment in four language pairs: English ⇔ German and English ⇔ Hindi. This resulted in improved BLEU and COMET scores in a low-resource setting.

1 Introduction

Training Neural Machine Translation (NMT) (Sutskever, 2014; Bahdanau, 2014; Luong, 2015; Vaswani, 2017) requires a large number of parallel corpora (Koehn and Knowles, 2017) and careful hyperparameter tuning (Sennrich and Zhang, 2019). Low-Resource Language (LRL) pairs generally possess a relatively limited amount of parallel data. In order to address the data scarcity problem, a possible solution is to utilize monolingual corpora (Wu et al., 2019). Using monolingual data, techniques such as generating synthetic parallel data via prompting Large Language Model (LLM) (Li et al., 2024; Enis and Hopkins, 2024), data augmentation via back translation (Hoang et al., 2018), Language Model (LM) prior (Baziotis et al., 2020), Knowledge Distillation (KD) or feature fusion using BERT (Yang et al., 2020; Zhu et al., 2020) and fine-tuning mBART (Zheng et al., 2021; San et al., 2024) have demonstrated a notable degree of performance improvement. But these approaches require training or fine-tuning of an additional teacher-like model to acquire text generation and translation capabilities or generate parallel corpora, followed by the transfer of knowledge to the Machine Translation (MT) model. However, recently available LLMs such as Llama (Dubey et al., 2024) have demonstrated remarkable proficiency in the translation task, which can be used to guide the MT model.

LLMs for translation (Hendy et al., 2023; Peng et al., 2023; Jiao et al., 2023) have shown significant success in generating high-quality translations. The deployment of these LLMs incurs substantial computational costs. LMs have been used in NMT to rerank the predictions of the MT model, or as an additional context, via LM fusion (Stahlberg et al., 2018), but lead to computational overhead, since LM is required during inference. Baziotis et al. (2020) proposed adding LM only in training and not in inference as a regularization term. However, this approach does not incorporate the source language information into LM when determining the regularization term, thereby failing to fully leverage the effectiveness of LLM.

We propose a new regularization term with the source sentence included to provide more context and replace LM with LLM to use its translation capabilities. Our contributions are as follows: (i) To the best of our knowledge, this is the first approach to using an instruction-tuned LLM as a regularization term, as described in Section 3 where both the source and target sentences are provided to the LLM as translation prompts. (ii) We evaluated the effects of using LLM in a low-resource setting and obtained an improvement in four directions: English-German (EN-DE), German-English (DE-EN), English-Hindi (EN-HI) and Hindi-English (HI-EN) (Section 4.5). In addition, we show that the proposed LLM prior outperforms the LM prior and baseline models.

2 Related Work

Baziotis et al. (2020) put the LM out of the MT model and the LM is used as a prior over the MT

Proceedings of the IJCNLP-AACL 2025 Student Research Workshop, pages 184–192
December 20-24, 2025

model's decoder by implementing posterior regularization using the loss function (Ganchev et al., 2010) in Equation 1:

$$\mathcal{L} = \sum_{t=1}^{N} - \log\ p_{\mathrm{MT}}(y_t | y_{<t}, \boldsymbol{x}) + \lambda\tau^2 \times \\ D_{\mathrm{KL}}(p_{\mathrm{MT}}(y_t | y_{<t}, \boldsymbol{x}; \tau)\ ||\ p_{\mathrm{LM}}(y_t | y_{<t}; \tau)), \tag{1}$$

where D_{KL}, $\boldsymbol{x}$ and y represent the Kullback–Leibler divergence, the source sentence and the target sentence, respectively, and $\boldsymbol{y} = y_1 y_2 ... y_N$. The posterior regularization includes prior information by imposing soft constraints on a posterior distribution of MT model. For computing D_{KL} between the MT model and LM distributions, softmax temperature parameters $\tau \geq 1$ are used. The same value of τ is applied to both LM and MT model at the same time. τ controls the smoothness of the output distributions $p_i = \frac{\exp(s_i/\tau)}{\sum_j \exp(s_j/\tau)}$, where s_i refers to the score (i.e., logit) obtained from the model before normalization of each word ID i. The magnitude of D_{KL} is on scales of $1/\tau^2$, so it is necessary to multiply its output by τ^2 to make the scale of D_{KL} loss invariant.

3 Proposed Approach

We propose using instruction-tuned LLM with source $\boldsymbol{x}$ to provide additional knowledge about the source language.

3.1 Loss Function

We changed p_{LM} of the loss function in Equation 1 with p_{LLM} and added the source $\boldsymbol{x}$ to it, resulting in the following equation.

$$\mathcal{L} = \sum_{t=1}^{N} - \log\ p_{\mathrm{MT}}(y_t | y_{<t}, \boldsymbol{x}) + \lambda\tau^2 \times \\ D_{\mathrm{KL}}(p_{\mathrm{MT}}(y_t | y_{<t}, \boldsymbol{x}; \tau)\ ||\ p_{\mathrm{LLM}}(y_t | y_{<t}, \boldsymbol{x}; \tau)), \tag{2}$$

where p_{LLM} is the probability distribution of the LLM conditioned on the translation prompt as in Figure 2. In Equation 2, the first term is the standard translation objective $\mathcal{L}_{\mathrm{MT}}$. The second term is the regularization term $\mathcal{L}_{\mathrm{KL}}$ referred to as the Kullback-Leibler divergence between the target side distributions of the MT model and the LLM output, weighted by λ. p_{LLM} can be viewed as weakly informative prior to the MT model distributions p_{MT}. It conveys partial information about $\boldsymbol{y}$. The LLM is

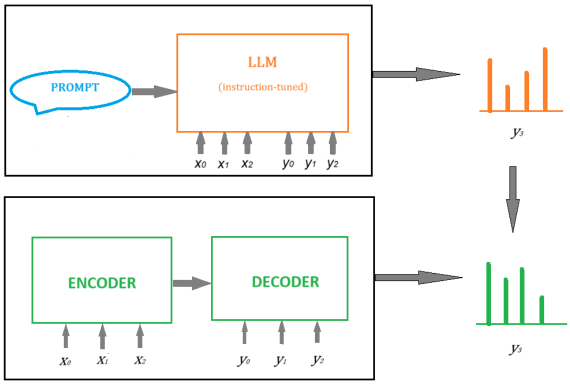

Figure 1: Distilling knowledge from LLM to MT model

```
prompt = [{ "role": "user",
        "content": "Translate the following from src_lang to
              tgt_lang: 'x'"},
      { "role": "assistant",
       "content": "\n\nThe translation of the sentence \"x\"
        from src_lang to tgt_lang is: \n\n\"y\""}]
```

Figure 2: Translation prompt used

no longer a component of the MT model architecture, and inference is conducted exclusively using the MT model.

3.2 Relation to Knowledge Distillation

The regularization term present in Equation 2 signifies the use of KD where the output probabilities of a larger teacher model are used to train a small student model as illustrated in Figure 1, minimizing D_{KL}. In standard KD (Hinton, 2015; Ba and Caruana, 2014; Buciluǎ et al., 2006), the teacher model is required to be trained with the same task as the student model, such as KD for machine translation (Kim and Rush, 2016) and KD for LLM (Gu et al., 2023; Ko et al., 2024; Agarwal et al., 2024; Zhong et al., 2024). These KD approaches can be of LogitKD (Hinton, 2015; Tan et al., 2019; Gu et al., 2023; Ko et al., 2024; Agarwal et al., 2024; Zhong et al., 2024), which optimizes the student model to minimize the difference between its predictions and the predicted distribution of the teacher model and of sequence KD (SeqKD) (Kim and Rush, 2016; Wang et al., 2021; Li et al., 2024), in which the student model learns from a synthetic target sequence generated by the teacher model. The SeqKD approach requires the generation of large amounts of synthetic data, which might require additional large-scale monolingual data. Therefore, our method is based on LogitKD and uses an LLM as the teacher model and an MT model as the student model. sq

4 Experimental Setup

Zhu et al. (2024) have shown that current LLM are more effective in machine translation from XX to EN than from EN to XX. To use LLM as a teacher model, we opt for Llama3.2[1] with a vocabulary size of 128,256, which is publicly available and supports eight languages with parameter sizes of 1B and 3B. We then evaluated the effectiveness of the LLM in situations where the amount of available parallel data is limited for the languages it supports. Therefore, we also conducted evaluation experiments using the EN–DE and EN–HI language pairs supported by Llama3.2-1B and Llama3.2-3B.

4.1 Training Data

275K and 188K bitexts were collected in EN-DE and EN-HI, respectively. These were then also formatted into DE-EN and HI-EN directions. Taking into account these bitext counts and following Koishekenov et al. (2023); Costa-Jussà et al. (2022)[2], we assumed that the language pairs are low-resource as they have between 100K and 1M bitexts. Also, we randomly sampled 10K bitexts to perform the experiment in a very low resource setting. EN-DE was acquired from WMT18 News Commentary v13[3], EN-HI was acquired from Opus WikiMatrix v2[4]. The official WMT-2017 test set and the FLORES-200[5] dev set were used as the validation set, and the WMT 2018 test set and FLORES-200 devtest set were used as the test set for EN-DE and EN-HI respectively. Monolingual data sets containing 3M and 30M sentences for each language were collected. The data sets prepared by Baziotis et al. (2020) were used to train English and German LM, and the News Crawls 2024[6] dataset was used to train Hindi LM.

4.2 Pre-processing

Fairseq[7] was used to train all models. For source languages, the sentencepiece (Kudo and Richardson, 2018)[8] tokenizer was used to train the tokenizer with a vocabulary size of 16,000. To distill the knowledge of the Llama3.2 model on the decoder side of the MT model, the MT model and Llama3.2 must share the same vocabulary and output space. Therefore, for target languages, we used the Llama3.2 model AutoTokenizer from the Transformers library (Wolf et al., 2020)[9]. With Fairseq, the final vocabulary 16,000 was generated for the encoder and 128,260 was generated for the decoder of the MT model which includes four additional specials tokens <s>, </pad>, </s> and <unk>.

4.3 Model Configuration

MT models are the Transformer architecture (Vaswani, 2017). LMs have a decoder layer only as shown in the Appendix A. We used the pre-trained and instruction-tuned Llama3.2 models with the default settings, employing the AutoModelForCausalLM class from the Transformers library. At each training step, the target sentence $\boldsymbol{y}$ in the case of the pre-trained or the translation prompt in Figure 2 in the case of the instruction tuned is passed as input to the AutoModelForCausalLM object to obtain the LLM probability distribution. For optimization, the Adam optimizer was used with a learning rate of 0.0005. The batch size was 32 sentences and 50 epochs with patience limit up to 10 epochs; that is, if the validation loss does not update for 10 consecutive validation epochs, the training stops. We extended Baziotis et al. (2020) implementation of using LM prior[10] to LLM prior.

4.4 Training and Inference

Approaches used to train MT models:

- **LM-KD** (Baziotis et al., 2020): defined in Equation 1.
- **LLM-KD** our comparison method: replaced p_{LM} by p_{LLM} defined in Equation 1.
- **LLM-Ins-KD** our proposed method: defined in Equation 2.

The training server specification is defined in Appendix A. LM (142M-3M text) and LM (142M-30M text) were trained for the English, German, and Hindi languages with 3 million and 30 million sentences, respectively. The MT model "LLM-KD (1B)" in EN-DE with different values of λ and τ was trained and calculated the BLEU scores on the validation data set. We found that the best values were $\lambda = 0.5$ and $\tau = 2$, as indicated in Appendix A. These hyperparameter values were used during

[1]https://huggingface.co/collections/meta-llama/llama-32
[2]https://github.com/nllb/train-example-count
[3]https://www.statmt.org/wmt18/translation-task.html
[4]https://opus.nlpl.eu/WikiMatrix
[5]https://github.com/openlanguagedata/flores
[6]https://data.statmt.org/news-crawl/hi/
[7]https://github.com/facebookresearch/fairseq
[8]https://github.com/google/sentencepiece
[9]https://github.com/huggingface/transformers
[10]https://github.com/cbaziotis/lm-prior-for-nmt

model	EN-DE		DE-EN		EN-HI		HI-EN	
	BLEU	**COMET**	**BLEU**	**COMET**	**BLEU**	**COMET**	**BLEU**	**COMET**
10K train set								
base (118M)	2.0	0.3080	1.8	0.3611	1.3	0.3380	0.8	0.3972
LM-KD (142M-3M text)	3.9	0.3509	4.8	0.4011	1.5	0.3487	1.0	0.4029
LM-KD (142M-30M text)	3.8	0.3674	4.7	0.4024	1.7	0.3485	0.7	0.3961
LLM-KD (1B)	4.1	0.3682	3.0	0.3786	1.1	0.3393	0.9	0.4061
LLM-KD (3B)	4.2	0.3668	2.8	0.3776	1.5	0.3478	1.0	0.4055
LLM-Ins-KD (1B-Ins) (ours)	**5.2**	**0.3771**	**5.9**	**0.4376**	**1.8**	**0.3778**	**1.4**	**0.4198**
LLM-Ins-KD (3B-Ins) (ours)	4.1	0.3651	4.9	0.4156	1.6	**0.3779**	**1.4**	**0.4240**
full train set								
base (118M)	23.8	0.6703	24.3	0.6850	14.9	0.6042	12.7	0.6690
LM-KD (142M-3M text)	24.8	0.6894	24.7	0.6876	14.7	0.5803	15.0	0.6930
LM-KD (142M-30M text)	25.6	0.6953	26.9	0.7209	14.6	0.5914	15.6	0.7043
LLM-KD (1B)	25.9	0.7014	26.9	0.7256	15.2	0.5937	15.3	0.7027
LLM-KD (3B)	25.7	0.7044	27.0	0.7254	16.3	0.6053	15.3	0.7011
LLM-Ins-KD (1B-Ins) (ours)	**27.6**	**0.7240**	**28.8**	**0.7457**	**16.7**	**0.6195**	**17.3**	**0.7251**
LLM-Ins-KD (3B-Ins) (ours)	27.3	0.7189	28.7	0.7418	16.3	0.6188	17.1	0.7242
prompting								
1B-Ins	17.0	0.6925	25.5	0.7887	6.3	0.5517	13.6	0.7500
3B-Ins	23.0	0.7765	33.1	0.8291	12.7	0.6317	20.6	0.7880

Table 1: Comparison of BLEU and COMET scores of each MT model on test data-set. Bold scores denote highest gain score in each section.

training. The trained MT models were used to translate the test data set. In addition, we prepared script to automatically obtain the translation output of Llama3.2 Instruct models by prompting with the same prompt mentioned in Figure 2 without the target sentence $\boldsymbol{y}$ and temperature $= 1$. The translations obtained were detokenized and converted into sentences. We calculated the BLEU scores using SacreBLEU (Post, 2018)[11] with default tokenizer "13a" and the COMET scores (Rei et al., 2020)[12] with "Unbabel/wmt22-comet-da".

4.5 Results

Table 1 shows the experimental results. For reference, we have included some translation examples in Appendix A. We also present BLEU and COMET scores for the teacher model in the bottom section of Table 1.

As indicated in bold letter, the MT model "LLM-Ins-KD (1B-Ins)" yielded an improvement in the BLEU score as well as the COMET score across all language pairs compared to all models. Training each LM took approximately five days using 4 GPUs. However, using the pre-trained Llama3.2 model, no training is required. This suggests that using an instruction-tuned LLM rather than an LM for KD to an MT model is more effective, provides enriched translation, and yields better results.

The instruction-tuned LLM outperformed the pre-trained LLM. This corroborates our hypothesis that pretrained LLM has better text generation capabilities but is unaware of the source sentence, which can mislead the target side of the MT model due to which the LLM-KD approach has not resulted in improvement in few language pairs than LM-KD.

Training the "LLM-Ins-KD (3B-Ins)" model did not result in higher BLEU or COMET scores than the "LLM-Ins-KD (1B-Ins)" model. However, the scores were approximately the same, as shown in Table 1. We hypothesize that the scores did not improve further due to the small capacity of the student MT model used. Significant differences in the capacity of the teacher and student models can affect performance, as discussed in (Cho and Hariharan, 2019; Fan et al., 2024).

"LLM-Ins-KD (1B-Ins)" MT model scores are close to those of the teacher models. This shows that "LLM-Ins-KD" leads to effective learning, but has room for further improvement. Teacher models have up to 3B parameters, but our trained MT models only have 118M, as indicated in Appendix A, so we achieved 96 % reduction in parameters.

Since Llama3.2 models have 1B or 3B parameters, it takes little more time and memory to provide logits for the KD process. So, the training time for the LLM-Ins-KD and LLM-KD methods was 1.5 times that of the LM-KD method. Our hypothesis is that the training time cost can be reduced by storing LLM in memory that we leave for future work.

[11] https://github.com/mjpost/sacrebleu
[12] https://github.com/Unbabel/COMET

5 Conclusion

In this work, we proposed knowledge distillation from a pre-trained LLM to a NMT model. We used both the text generation and translation capabilities of the LLM. This approach is suitable because we do not need any monolingual data set or additional teacher model training. We also achieved improvement in BLEU and COMET scores for all language pairs compared to baselines in a low resource setting. We demonstrated that using the instruction-tuned LLM can be more effective than using the LM to distill knowledge to MT model.

Limitations

First, we used the lightweight open-source Llama 3.2 1B and 3B models for our experiment. We could have chosen larger LLMs, such as 8B or 70B, but we opted for the smaller models to perform the experiment quickly and with less computational cost. Second, we compared the BLEU and COMET scores of the translation model with the Llama3.2-1B-Instruct model. LLM return a translation output with extra description when inference is made with translation prompts. To automatically extract only the translation sentences, we wrote a program script. However, we believe that this approach might not be suitable. There may be a better way to obtain only the translated output from the LLM inference pipeline.

Acknowledgements

These research results were obtained from the commissioned research (No.22501) by National Institute of Information and Communications Technology (NICT), Japan. This work was supported by JSPS KAKENHI Grant Number JP24K15071.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27.

Dzmitry Bahdanau. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Christos Baziotis, Barry Haddow, and Alexandra Birch. 2020. Language model prior for low-resource neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7622–7634, Online. Association for Computational Linguistics.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Jang Hyun Cho and Bharath Hariharan. 2019. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Maxim Enis and Mark Hopkins. 2024. From LLM to NMT: Advancing low-resource machine translation with claude. *arXiv preprint arXiv:2404.13813*.

Wen-Shu Fan, Xin-Chun Li, and De-Chuan Zhan. 2024. Exploring dark knowledge under various teacher capacities and addressing capacity mismatch. *arXiv preprint arXiv:2405.13078*.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Amr Hendy, 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. *arXiv preprint arXiv:2302.09210*.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *2nd Workshop on Neural Machine Translation and Generation*, pages 18–24. Association for Computational Linguistics.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is ChatGPT a good translator? yes with GPT-4 as the engine. *arXiv preprint arXiv:2301.08745*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. DistiLLM: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Yeskendir Koishekenov, Alexandre Berard, and Vassilina Nikoulina. 2023. Memory-efficient NLLB-200: Language-specific expert pruning of a massively multilingual machine translation model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3567–3585, Toronto, Canada. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jiahuan Li, Shanbo Cheng, Shujian Huang, and Jiajun Chen. 2024. MT-PATCHER: Selective and extendable knowledge distillation from large language models for machine translation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6445–6459, Mexico City, Mexico. Association for Computational Linguistics.

Minh-Thang Luong. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of ChatGPT for machine translation. *arXiv preprint arXiv:2303.13780*.

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

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 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.

Mya Ei San, Sasiporn Usanavasin, Ye Kyaw Thu, and Manabu Okumura. 2024. A study for enhancing low-resource Thai-Myanmar-English neural machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(4):1–24.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. *arXiv preprint arXiv:1905.11901*.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. *arXiv preprint arXiv:1809.00125*.

I Sutskever. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. In *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 6456–6466, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *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 4207–4216.

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

Francis Zheng, Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. 2021. Low-resource machine translation using cross-lingual language model pretraining. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 234–240.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. 2024. Revisiting knowledge distillation for autoregressive language models. *arXiv preprint arXiv:2402.11890*.

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

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

A Appendix

A.1 Architecture of the Models

Table 2 shows the architecture of the different models used in these experiments along with the number of parameters.

component	**value**			
	MT	**LM**	**1B**	**3B**
parameters	118M	142M	1B	3B
Embedding	512	1024	2048	3072
Encoder layer	6	6	N/A	N/A
Decoder layer	6	6	16	28
Encoder head	8	8	N/A	N/A
Decoder head	8	16	N/A	N/A
Dropout (all)	0.3	0.3	N/A	N/A

Table 2: Architecture of each model used

A.2 Specification of Training Server

The specification of the training server for this experiment is shown in Table 3.

hardware	**capacity**
GPU	47GB
number of GPU	1-4
CPU	6-8 core
RAM	40-60 GB
total training time	15days

Table 3: Specification of training server

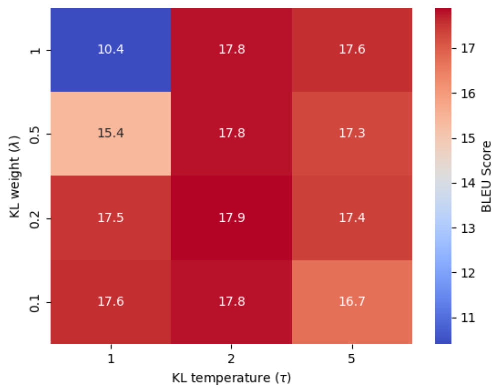

Figure 3: Valid set BLEU scores of "LLM-KD (1B)" in the EN-DE direction with different value of λ and τ

A.3 Hyperparameter Tuning

Figure 3 shows the heat map of the valid-set BLEU scores with different combinations of λ and τ in the EN-DE direction. This MT model trained with our comparison method: replaced p_{LM} by p_{LLM} defined in Equation 1.

Taking the baseline BLEU score of the MT model 16.8, we see the pattern as follows: Using $\tau = 2$ results in the MT model to acquire more dark knowledge encoded in the LLM logits, and at this stage, changing λ affects the performance of the MT model. So, we selected $\lambda = 0.5$ and $\tau = 2$ to train all models in our experiments.

A.4 Translation Examples

Table 7 provides some translation examples.

Source	Munich 1856: Four maps that will change your view of the city
Reference	München 1856: Vier Karten, die Ihren Blick auf die Stadt verändern
Trained with 10K train set	
base (118M)	Mushalt 2015: 2006 wird das Bürgerkrieg gegenüber den Vereinigten Staaten erwartet werden.
LM-KD (142M-3M text)	München: Im Jahr 1865 wird die Stadtkarte auf den Inseln gestoppt werden.
LM-KD (142M-30M text)	München: Im Jahr 1865 wird die Stadt von den Inseln gestohlen, um die Stadt zu den Inseln zu verfehlen.
LLM-KD (1B)	Menschen 1865 wird das Begrüßte angesichts der Stadt veränderten, dass die Stadtveränderung der Stadt erkannt werden.
LLM-KD (3B)	Menschen wird 1862 verfügt: Die Befürdigen der Stadt verändert werden.
LLM-Ins-KD (1B-Ins) (ours)	München 18. Dezember 1861 verfügt: Die Hoffnung der Stadt erfüllt.
LLM-Ins-KD (3B-Ins) (ours)	Menschen 1866 wird 1861 ein Südenkrieg beigetragen: Der Bürger der Stadt ziehen.
Trained with full train set	
base (118M)	München 1856: Vier Landkarten, die Ihre Sichtweise der Stadt ändern werden
LM-KD (142M-3M text)	München 1856: Vier Landkarten, die Ihre Sichtweise der Stadt ändern werden
LM-KD (142M-30M text)	München 1856: Vier Karten, die Ihre Sicht der Stadt ändern werden.
LLM-KD (1B)	München 1856: Vier Landkarten, die Ihre Sicht der Stadt verändern werden.
LLM-KD (3B)	München 1856: Vier Landkarten, die Ihre Sicht der Stadt verändern werden
LLM-Ins-KD (1B-Ins) (ours)	München 1856: Vier Karten werden Ihre Sicht der Stadt ändern
LLM-Ins-KD (3B-Ins) (ours)	München 1856: Vier Landkarten, die Ihre Ansicht in der Stadt verändern werden.

Table 4: EN-DE translation example

Source	München 1856: Vier Karten, die Ihren Blick auf die Stadt verändern
Reference	Munich 1856: Four maps that will change your view of the city
Trained with 10K train set	
base (118M)	meanwhile, 6.6% of your city are on the city of your city.
LM-KD (142M-3M text)	meanwhile, 6.6% of your city are on the city of your city.
LM-KD (142M-30M text)	after all, 6.6% of your books, you are changing the city of your city.
LLM-KD (1B)	the 1 of 1, 1,000 deaths on the city of the city.
LLM-KD (3B)	every year, 1, 1,000 I am on the city of the city of the city.
LLM-Ins-KD (1B-Ins) (ours)	Abba 1876,000 met the city of your city on your city to change.
LLM-Ins-KD (3B-Ins) (ours)	Copenhagen 18,000 die at the city of the city of the city to leave the city of the city.
Trained with full train set	
base (118M)	Munich 1856: Four Crises changing your eyes on the city
LM-KD (142M-3M text)	Munich, 1856: Four maps changing your eyes to the city
LM-KD (142M-30M text)	Munich, 1856: Four cards change your view of the city.
LLM-KD (1B)	Munich, 1856: Four maps changing your eyes to the city
LLM-KD (3B)	Munich, 1856: Four maps changing your eyes to the city
LLM-Ins-KD (1B-Ins) (ours)	Munich 1856: Four maps changing your eyes on the city
LLM-Ins-KD (3B-Ins) (ours)	Munich 1856: Four cards that change your eyes on the city

Table 5: DE-EN translation example

Source	”While one experimental vaccine appears able to reduce Ebola mortality
Reference	”जबकि एक प्रायोगिक वैक्सीन इबोला से मृत्यु दर में कमी हो सकती है
Trained with 10K train set	
base (118M)	” इस प्रकार के लिए बहुत कम हो जाता है।
LM-KD (142M-3M text)	”प्रत्येक व्यक्ति को कवर करने के लिए एक मूर्ति का प्रयास किया गया है।
LM-KD (142M-30M text)	”जो मात्मा को माता है कि मात्मा को मात्मा मिल जाता है।
LLM-KD (1B)	हालांकि का पूरा है।
LLM-KD (3B)	इसी मृत्यु का प्रयोग किया जा सकता है।
LLM-Ins-KD (1B-Ins) (ours)	बूगल को मृत्यु के लिए एक सप्ताह में खोला जाता है।
LLM-Ins-KD (3B-Ins) (ours)	” फिल्म का प्रयोग किया जाता है।
Trained with full train set	
base (118M)	”व्हील एक प्रयोगात्मक टीका ईबोला मृत्यु को कम करने में सक्षम है।
LM-KD (142M-3M text)	एक प्रयोगात्मक टीका इबोला मृत्यु दर कम करने में सक्षम होता है।
LM-KD (142M-30M text)	एक प्रायोगिक टीका एबोला मृत्यु दर को कम करने में सक्षम होता है।
LLM-KD (1B)	एक प्रयोगात्मक टीका ईबोला मृत्यु दर को कम करने में सक्षम है।
LLM-KD (3B)	एक प्रयोगात्मक वैक्सीन एबोला मृत्यु को कम करने में सक्षम दिखाई देता है।
LLM-Ins-KD (1B-Ins) (ours)	एक प्रयोगात्मक टीका मृत्यु मृत्यु को कम करने में सक्षम है।
LLM-Ins-KD (3B-Ins) (ours)	”एक प्रयोगात्मक वैक्सीन एबोला मृत्यु को कम करने में सक्षम लगता है।

Table 6: EN-HI translation example

Source	”जबकि एक प्रायोगिक वैक्सीन इबोला से मृत्यु दर में कमी हो सकती है
Reference	”While one experimental vaccine appears able to reduce Ebola mortality
Trained with 10K train set	
base (118M)	It is one of them to make it it it it to be in 10.
LM-KD (142M-3M text)	It can be one of an important time, but it can be used for a time.
LM-KD (142M-30M text)	It can be an important for an time.
LLM-KD (1B)	It is one of one of it is one person to be a matter of it.
LLM-KD (3B)	It is one of a person to be one of one person to be a matter of people.
LLM-Ins-KD (1B-Ins) (ours)	It can be one of one of a person to be about 1000.
LLM-Ins-KD (3B-Ins) (ours)	It is one of one of an reason, but is one of about 1000.
Trained with full train set	
base (118M)	As a result, an active vaccine may be decreased in the rate of death.
LM-KD (142M-3M text)	”Exposure to an experimental vaccine may reduce mortality rates from an outbreak.
LM-KD (142M-30M text)	”Currently an experimental vaccine may be reduced to death rates”.
LLM-KD (1B)	”While one experimental vaccine appears able to reduce Ebola mortality
LLM-KD (3B)	”An experimental vaccine may be reduced to death rates”.
LLM-Ins-KD (1B-Ins) (ours)	”Failure to reduce mortality rate by immunoscopy”.
LLM-Ins-KD (3B-Ins) (ours)	A pilot vaccine may reduce mortality rate from the immunoglobulin.

Table 7: HI-EN translation example