The Journal of Open Source Software

ALF-T5: Adaptive Neural Machine Translation Framework for Constructed and Natural Languages

Matheus J. G. Silva [1¶]

1 Anhanguera University, Brazil ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 09 October 2025
Published: unpublished

Summary

Neural machine translation (NMT) typically demands abundant parallel corpora, restricting its reach to widely spoken languages and limiting exploration of low-resource and constructed languages (conlangs). **ALF-T5** is a Python framework that enables **few-shot**, **bidirectional** translation between English and target languages, natural or constructed, by leveraging transfer learning, parameter-efficient fine-tuning (LoRA via PEFT), and data augmentation. It supports interactive usage, batch translation, automatic evaluation (BLEU, METEOR), and architectural experimentation, making it useful for research, teaching, and prototype development.

Statement of need

Most modern NMT toolkits assume large parallel corpora and substantial compute. This restricts experimentation with under-resourced languages and constructed languages, and limits exploration of translation strategies in extremely low-data regimes.

ALF-T5 fills this gap by enabling translation model training from as few as **10–50 example pairs**. The project emphasizes usability (CLI + Python API), reproducibility (built-in metrics and tests), and modularity (pluggable augmentation and experiment components). Because it integrates with the widely used Transformers and PEFT ecosystems, ALF-T5 is usable on modest hardware and useful for both educators and researchers.

Technical approach

Model and loss formulation

Let the source sequence be $X = (x_1, x_2, \dots, x_n)$ and the target sequence be $Y = (y_1, y_2, \dots, y_m)$. The model factorizes the conditional probability as

$$P(Y \mid X) = \prod_{t=1}^{m} P(y_t \mid y_{1:t-1}, X).$$

We train using the standard cross-entropy loss:

$$\mathcal{L}_{\text{CE}}(X, Y) = -\sum_{t=1}^{m} \log P(y_t \mid y_{1:t-1}, X).$$

To support bidirectional translation, we train both directions and combine losses:

$$\mathcal{L}_{\text{bi}} = \mathcal{L}_{\text{CE}}(X \to Y) + \mathcal{L}_{\text{CE}}(Y \to X).$$

(Using $y_{1:t-1}$ instead of $y_{<t}$ avoids characters that can be converted or escaped by the toolchain.)

Parameter-efficient tuning (LoRA / PEFT)

ALF-T5 uses LoRA (low-rank adaptation) implemented through the PEFT framework: the pretrained base model weights remain frozen and we learn small low-rank adapter matrices. This drastically reduces the number of trainable parameters and lowers memory / compute requirements compared to full fine-tuning (Hu et al., 2021). The adapters are inserted at attention / projection layers and trained while the backbone stays unchanged.

Data augmentation

To amplify scarce parallel data we use lightweight augmentations:

- Case variation (lowercase / uppercase / title)
- Word-order permutations within short phrases
- Vocabulary recombination (mixing subword segments or analogous tokens)

These synthetic variants increase the effective dataset size and help generalization in low-data settings.

Usage & features

- **Interactive mode**: CLI REPL for single-sentence translation with optional confidence scores.
- **Batch mode**: File and list translation for large-scale processing.
- **Automatic evaluation**: BLEU and METEOR computed during or after training.
- **Experiment framework**: Grid search over hyperparameters (LoRA rank, learning rate) with result logging.
- **Model inference**: Load a saved translator and translate new text via a simple API.

Citations

This project builds on a number of foundational works and libraries, including T5 (Raffel et al., 2020), the Transformer architecture (Vaswani et al., 2017), LoRA (Hu et al., 2021), BLEU (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), and the Hugging Face Transformers library (Wolf et al., 2020).

Figures

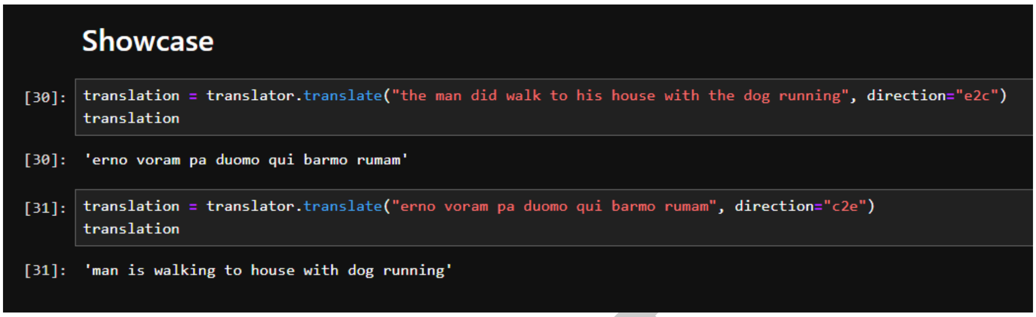

Figure 1: Few-shot translation example in a constructed language (45 training pairs).

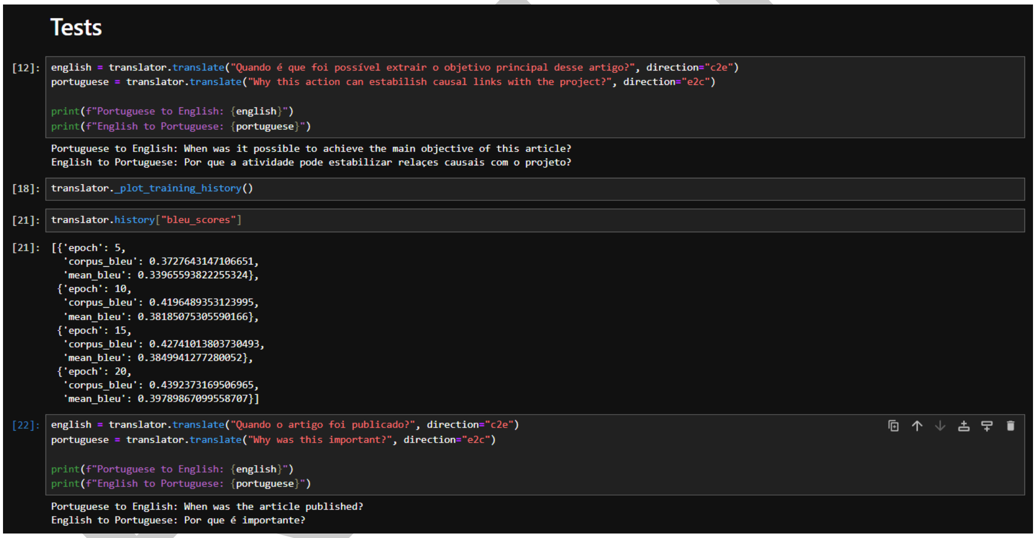

Figure 2: Portuguese-English translation performance (BLEU / METEOR).

Acknowledgements

Thanks to the authors of T5 and LoRA, and to the Hugging Face and PEFT teams. Special thanks to Paulo Pirozelli for providing the Pirá dataset used in evaluation, and to early testers who provided feedback.

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72. https://aclanthology.org/W05-0909.pdf

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. *arXiv Preprint arXiv:2106.09685*. https://arxiv.org/abs/2106.09685

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu,

P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67. http://jmlr.org/papers/v21/20-074.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Łukasz, & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*. https://arxiv.org/abs/1706.03762

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P. von, Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., … Rush, A. M. (2020). *Transformers: State-of-the-art natural language processing*. https://arxiv.org/abs/1910.03771