Adapting Mammoth Library for Sign Language Translation

Abstract

Multilingual sign language translation is gaining increased attention due to its potential to bridge communication gaps across diverse linguistic communities. In this work, we adapt the MAMMOTH toolkit for bilingual sign language translation by training it on the Phoenix2014T dataset, which focuses on German Sign Language (DGS). MAMMOTH, a modular Neural Machine Translation (mNMT) system, enables flexible parameter sharing across components like word embeddings, encoder states, and attention mechanisms. By leveraging its efficient GPU allocation strategies, we optimized hardware usage, reducing data transfer and enhancing parallel processing. This adaptation highlights the potential of modular architectures to scale and support future advancements in sign language translation.

1 Introduction

Scaling multilingual Neural Machine Translation (NMT) models to accommodate a large number of languages often results in performance degradation due to a phenomenon known as the "curse of multilinguality." This issue arises when a model's limited capacity is stretched across multiple languages, leading to interference that negatively impacts per-language performance [1, 2, 3]. Similar challenges are anticipated in the realm of multilingual sign language translation, especially with the emergence of new large-scale datasets such as JWSign [4] and YouTube-SL-25 [5]. As these datasets cover multiple sign languages, the same interference effects are expected to manifest, complicating efforts to scale models effectively.

In this paper, we focus on a more constrained but equally important challenge: adapting the massively multilingual modular open translation (MAMMOTH) NMT library for bilingual sign language translation (BSLT). While MAMMOTH is designed for modular and multilingual NMT, we explore its application to sign language translation, with the aim of leveraging its modular architecture to minimize interference between languages. This adaptation allows for more efficient parameter sharing and task-specific fine-tuning, offering a potential solution to the challenges of scaling sign language translation models.

To evaluate our approach, we train and test our method on the Phoenix2014T dataset [6], a widely-used benchmark for German Sign Language (DGS) translation. By focusing on bilingual translation, we aim to provide insights into how modular NMT toolkits like MAMMOTH can be adapted for sign language tasks and serve as a foundation for future research into multilingual sign language translation.[1]

[1] Our code is publicly available at `https://github.com/DFKI-SignLanguage/video-mammoth`

2 Related Work

While there are several open-source frameworks for training Neural Machine Translation (NMT) models, such as Fairseq [7], which supports modular components, MAMMOTH is the first toolkit explicitly designed for modular and multilingual NMT. MAMMOTH provides extensive flexibility in its modularity, allowing for dynamic parameter sharing and task-specific configurations, making it uniquely suited for multilingual tasks across different language pairs.

Another comparable framework is AdapterHub [8], which extends the Hugging Face Transformers library to support lightweight adapters for various tasks, including multilingual NMT. AdapterHub enables efficient model fine-tuning by introducing task-specific adapters without the need to retrain the entire model. Although similar in its modular approach, AdapterHub is not specialized for large-scale NMT tasks in the way MAMMOTH is, nor does it emphasize multilingualism to the same extent.

MAMMOTH builds upon the foundation of the OpenNMT-py toolkit [9], which is a modular NMT system known for its flexibility and efficiency in research and production environments. However, MAMMOTH extends these capabilities further by emphasizing parameter sharing and hardware-efficient multi-task learning, making it a highly specialized tool for complex multilingual NMT systems.

Sign language translation remains an underexplored area within this context. To date, there has been no direct application of MAMMOTH or similar modular frameworks to sign language translation tasks. The closest related work is Sign2GPT [10], which leverages pretrained vision and language models via lightweight adapters for gloss-free sign language translation. Unlike MAMMOTH, Sign2GPT does not rely on modular NMT components but rather utilizes adapters for efficient transfer learning in sign language tasks. This represents a parallel approach to efficient model adaptation, but without the comprehensive modularity that Mammoth offers.

3 Overview of MAMMOTH's Design

The MAMMOTH toolkit is structured around the concept of a *task*, which governs the models behavior and remains fixed throughout the entire training process. A task is defined by three core components:

1. **Set of Modules**: These modules represent the key components of the model, such as encoders and decoders, that are responsible for specific language processing tasks. In translation scenarios, the modules are assigned to handle particular language pairs. For instance, in a Swahili-to-Catalan translation task, the modules involved would focus on Swahili encoding and Catalan decoding.
2. **Preprocessing Steps**: Each task incorporates a uniform set of preprocessing procedures. These steps, such as tokenization, ensure that all input data is processed consistently across the entire dataset for that task.
3. **Dataset (Parallel Corpus)**: A task is associated with a single dataset, typically a parallel corpus (bitext), where aligned source and target language pairs are provided. This ensures that all data used in the task follows the same structure and configuration.

In translation tasks, the combination of modules, preprocessing steps, and dataset defines the model's behavior. Each data point must adhere to the defined task structure, using the same modules and preprocessing rules, and can be grouped into a single parallel corpus for efficient processing.

MAMMOTH further enforces that each task is assigned to a specific GPU or compute node. All relevant modules are hosted on this device, minimizing inter-device communication and maximizing computational efficiency. By localizing task-specific operations, MAMMOTH reduces overhead and optimizes resource usage, especially in multi-tasking environments.

Historically, MAMMOTH builds on the OpenNMT-py framework [9], extending its modularity to allow for flexible configuration and sharing of components across different tasks. This modular design enables MAMMOTH to support scalable and customizable neural machine translation (NMT) workflows.

4 Experimental Setting

In this section, we describe the dataset, pre-processing steps, and the evaluation metrics we use for training and testing the MAMMOTH library for bilingual sign language translation.

4.1 Dataset

For this work, we used the Phoenix2014T dataset [11], a large-scale collection of German Sign Language (DGS) videos. The dataset features interpreters translating weather forecasts, and includes gloss annotations as well as spoken German translations.

4.2 Data Processing

For the **video processing**, we use the sign features based on the spatial embedding approach introduced by authors in [12]. This method has been employed in various sign language translation works such as [6] and [13], proving effective for representing sign language in a continuous space.

For the **text processing**, we train a SentencePiece [14] tokenizer on the Phoenix2014T training set with a vocabulary size of 2000. This tokenizer provides subword-level segmentation, ensuring robust handling of rare words and facilitating better translation performance.

4.3 Evaluation Metrics

To evaluate the final model on the test set, we use the BLEU score [15], a standard metric for assessing the quality of machine translation outputs. Specifically, we use the sacreBLEU [2] implementation [16] to ensure consistency and comparability of the results.

5 Experimental Results

In this section, we present the results of our experiments using the adapted MAMMOTH framework for BSLT. Our primary objective was to train the system for optimal performance on the Phoenix2014T dataset, specifically focusing on improving the translation accuracy as measured by the BLEU score.

We experimented with various hyperparameters in an effort to improve model performance. The following configuration provided the best results:

- **Number of encoder and decoder layers:** 3 layers each
- **Optimizer:** Adam optimizer
- **Learning rate:** 0.005
- **Learning rate decay:** 0.5

Despite tuning these hyperparameters, the model was unable to achieve a BLEU score higher than **1.97**, with the following BLEU breakdown: **11.4/2.3/1.0/0.7** for 1-gram, 2-gram, 3-gram, and 4-gram precision, respectively. These results indicate the significant challenges associated with achieving accurate bilingual sign language translation using this architecture.

The qualitative results can be seen in Table 1, which compares the ground truth reference translations with the models predictions.

[2] `BLEU|nrefs:1|case:mixed|eff:yes|tok:13a|smooth:exp|version:1.4.22`

Reference Translation	**Model's Prediction**
sonst ein wechsel aus sonne und wolken	und nun die wettervorhersage für morgen dienstag den fünften januar
der wind weht schwach bis mäßig an der nordsee und im bergland auch frischer wind	und nun die wettervorhersage für morgen montag den fünfundzwanzigsten januar
heute nacht liegen die werte zwischen vierzehn und sieben grad	und nun die wettervorhersage für morgen samstag den zwölften september
am sonntag vor allem in der südosthälfte gewitterschauer sonst setzt sich wieder meist die sonne durch	am sonntag scheint häufig die sonne im südosten häufig die sonne
heute nacht neunzehn bis fünfzehn grad im südosten bis zwölf grad	heute nacht neunzehn bis fünfzehn grad im süden bis zwölf grad

Table 1: Ground Truth vs Model's Predictions

6 Conclusion

In this work, we focused on modular architecture for multilingual sign language translation, specifically adapting the MAMMOTH framework for bilingual sign language translation as our first step. While our efforts to train the model on the Phoenix2014T dataset were systematic, the results revealed that the model achieved a low BLEU score, highlighting the challenges associated with this task.

References

[1] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics.

[2] Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. On negative interference in multilingual models: Findings and a meta-learning treatment. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online, November 2020. Association for Computational Linguistics.

[3] Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. Lifting the curse of multilinguality by pre-training modular transformers. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States, July 2022. Association for Computational Linguistics.

[4] Shester Gueuwou, Sophie Siake, Colin Leong, and Mathias Müller. JWSign: A highly multilingual corpus of Bible translations for more diversity in sign language processing. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9907–9927, Singapore, December 2023. Association for Computational Linguistics.

[5] Garrett Tanzer and Biao Zhang. Youtube-sl-25: A large-scale, open-domain multilingual sign language parallel corpus, 2024.

[6] Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and R. Bowden. Neural sign language translation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018.

[7] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[8] Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. Adapters: A unified library for parameter-efficient and modular transfer learning. In Yansong Feng and Els Lefever, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160, Singapore, December 2023. Association for Computational Linguistics.

[9] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In Mohit Bansal and Heng Ji, editors, *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[10] Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. Sign2GPT: Leveraging large language models for gloss-free sign language translation. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-weather. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1911–1916, May 2014.

[12] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3416–3424, Honolulu, HI, USA, July 2017.

[13] Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and R. Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10020–10030, 2020.

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

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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. Association for Computational Linguistics, July 2002.

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