

sign.mt: Real-Time Multilingual Sign Language Translation Application

Amit Moryossef
amit@sign.mt

If you want to go fast, go alone;
If you want to go far, go together.

Abstract

This paper presents *sign.mt*, an open-source application for real-time multilingual bi-directional translation between spoken and signed languages. Harnessing state-of-the-art open-source models, this tool aims to address the communication divide between the hearing and the deaf, facilitating seamless translation in both spoken-to-signed and signed-to-spoken translation directions.

To provide reliable and unrestricted communication, *sign.mt* offers offline functionality, crucial in areas with limited internet connectivity. It enhances user engagement by providing customizable photorealistic sign language avatars, encouraging a more personalized and authentic user experience.

Licensed under [CC BY-NC-SA 4.0](#), *sign.mt* signifies an important stride towards open, inclusive communication. The app can be used and modified for personal and academic purposes and even supports a translation API, fostering integration into a wider range of applications. However, it is by no means a finished product.

We invite the NLP community to contribute towards the evolution of *sign.mt*. Whether it be the integration of more refined models, the development of innovative pipelines, or user experience improvements, your contributions can propel this project to new heights. Available at <https://sign.mt>, it stands as a testament to what we can achieve together, as we strive to make communication accessible to all.

1 Motivation

Sign language translation applications are crucial tools for enabling communication between individuals who are deaf or hard of hearing and those who communicate through spoken language. However, the complexity of developing sign language translation applications goes beyond handling mere text. These applications must be able to process

and generate videos, demanding additional considerations such as compute capabilities, accessibility, usability, handling large files, and platform support.

sign.mt, standing for **Sign Language Machine Translation**, was conceived as a response to these challenges. Current research in the field of sign language translation is fragmented and somewhat nebulous, with different research groups focusing on various aspects of the translation pipeline or specific languages. Moreover, the high costs associated with server-side deployment and the complexity of client-side implementations often deter the development of interactive demonstrations for newly proposed models.

By providing a comprehensive application infrastructure that integrates the essential features around the translation process, *sign.mt* serves as a dynamic proof-of-concept. It aims to streamline the integration of new research findings into the application, sidestepping the overhead typically associated with implementing a full-stack application. When a research group develops a new model or improves a pipeline, they can integrate their advancements into the app swiftly, focusing only on their model. This approach allows researchers to deploy the app in a branch, testing their models in a practical environment. If the license allows and the models show an improvement, they can contribute their models to the main codebase. This is the first tool of its kind, diverging significantly from closed-source commercial applications.

Further, *sign.mt* serves as a multilingual platform, thus unifying the fragmented research landscape. It enables the concurrent running of models from different research groups for the supported languages, providing users with state-of-the-art translation capabilities for each language. Through this, *sign.mt* not only enhances accessibility and communication but also fuels continuous innovation in sign language translation research.

2 Implementation

Sign language translation presents unique challenges that set it apart from text-based translation. While text-based translation operates entirely within the textual domain for both input and output, sign language translation involves cross-modal transformation – from text to video and vice versa. This demands distinct implementations not only in functionality but also in the user interface.

It is essential to emphasize that the specific models utilized within various pipelines are deliberately modular and interchangeable. Our current choice of models for each module or task is primarily opportunistic, driven by availability rather than performance metrics or user evaluations. The app serves as a dynamic orchestrator, seamlessly coordinating among these models to deliver an integrated user experience. The platform’s design accommodates the likelihood that researchers or users may wish to experiment with different models or fine-tune existing pipelines, without being constrained by rigid implementation details.

2.1 Spoken-to-Signed Translation

Through this pipeline (Figure 1), *sign.mt* is capable of real-time translation from spoken language audio (or text) into sign language video, further democratizing communication across modalities.

For spoken-to-signed translation, the process begins with an input of spoken language text. Optionally, we allow audio input, which is first transcribed into spoken language text using on-device Speech-to-Text (STT) technology.

When the input language is unknown, the text undergoes Spoken Language Identification (using MediaPipe (Lugaresi et al., 2019) or cld3 (Salcianu et al., 2016)), which detects the language of the provided text. This is crucial for choosing the appropriate model for subsequent translation steps. Simultaneously, the text is optionally normalized (using ChatGPT (OpenAI, 2022)). This includes fixing capitalization, punctuation, grammatical errors, or misspellings, which we have found to enhance the performance of subsequent translation stages. The language-identified and potentially normalized text is then split into individual sentences using the on-device internationalized segmentation service (Mozilla Developer Network, 2020).

Each sentence is then individually translated into SignWriting (Sutton, 1990). Here, our system leverages real-time client-side machine translation (Bo-

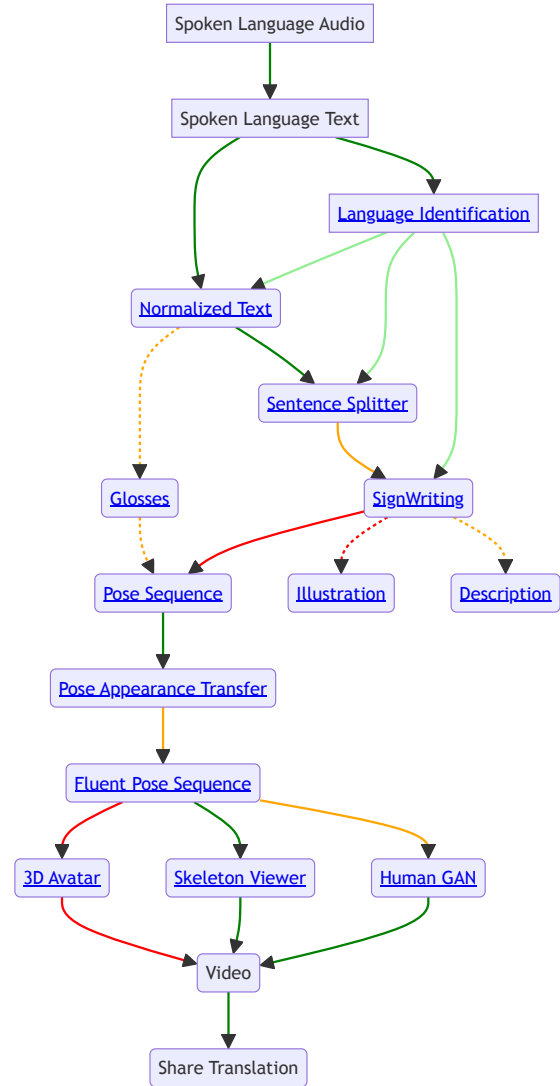


Figure 1: The Spoken-to-Signed translation pipeline.

goychev et al., 2021) to translate the grammatical structures and lexicon of spoken languages into the visual-gestural modality of sign languages (Jiang et al., 2023; Moryossef and Jiang, 2023).

The SignWriting output is then converted into a pose sequence (Inspired by Arkushin et al. (2023)), representing the signed sentence. After undergoing appearance transfer to always show the same person (Moryossef, 2024), this pose sequence is the input for the rendering engine, with three options: Skeleton Viewer (Minimalistic visualization of the skeletal pose (Moryossef and Müller, 2021)) Human GAN (Pix2Pix (Isola et al., 2017; Shi et al., 2016) image-to-image model, generating a realistic human avatar video), and a 3D Avatar (Neural model to translate between pose positions and rigged rotations, performing the signs).

These different outputs provide users with a choice on how they prefer to view the translation, catering to a broad range of preferences and use cases. The skeleton viewer is useful for developers to see the raw output, as well as for low-compute users. The 3D Avatar is useful in mixed reality applications, where it can be integrated into the environment, and the Human GAN is useful for high-compute users, facilitating a natural interaction.

Currently, while we don't have a fully functional SignWriting to pose animation model, we have created a baseline model as an interim solution (Moryossef et al., 2023b). This model performs dictionary-based translation from the spoken language text directly to poses, bypassing the SignWriting stage. However, it's important to note that there are numerous common cases in sign languages that this baseline model cannot handle adequately yet. We have made the baseline model open-source, and it is available for further improvements and contributions from the community at <https://github.com/sign-language-processing/spoken-to-signed-translation>. We hope that this open-source approach will stimulate further research and development in this area, allowing for the integration of more sophisticated and accurate models in future iterations of the application.

2.2 Signed-to-Spoken Translation

Through this pipeline (Figure 2), *sign.mt* can take a sign language video and output corresponding spoken language text or audio in real-time. The offline functionality of the app ensures that this feature remains accessible even in areas with limited connectivity, provided that the models are cached on the device.

For signed-to-spoken translation, the source is a video (either by the user uploading a pre-existing sign language video or using the camera to record a live sign language video). Our current pipeline takes the video, and using Mediapipe Holistic (Grishchenko and Bazarevsky, 2020) pose estimation extracts the full body pose from each frame.

This pose information is then fed into a Segmentation module (Moryossef et al., 2023a), which segments distinct signs within the continuous signing flow, as well as phrase boundaries. The segmented signs are subsequently lexically transcribed using SignWriting (Sutton, 1990), a comprehensive sys-

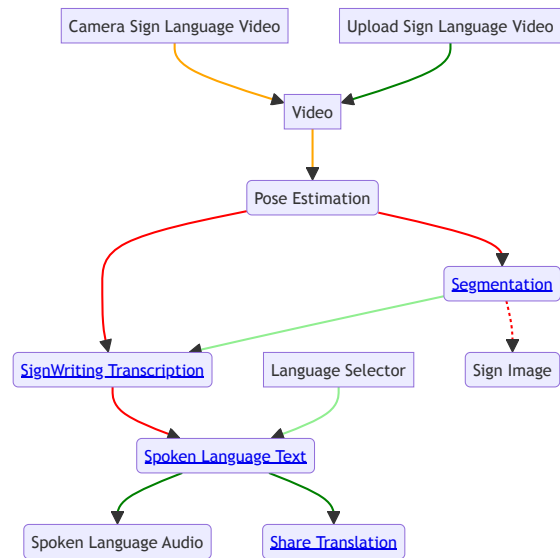


Figure 2: The Signed-to-Spoken translation pipeline.

tem for transcribing sign languages visually.

This SignWriting transcription serves as the textual input for the translation model, which translates it into corresponding spoken language text (Jiang et al., 2023; Moryossef and Jiang, 2023). This text is then optionally converted into spoken language audio using on-device Text-to-Speech (TTS), providing an auditory output for the user.

3 User Engagement

The impact of *sign.mt* can be measured by its widespread and consistent usage, highlighting the tremendous growth potential as the app continues to slowly improve.



Figure 3: Distribution of *sign.mt* users across the world, over the last year.

Figure 3 depicts the global adoption of *sign.mt*, with users distributed across multiple countries. None of these top user countries are home to the core developer of the app.

As shown in Figure 4, *sign.mt* demonstrates slow but consistent user growth (by Google Analytics), indicative of its reliability and sustained relevance.

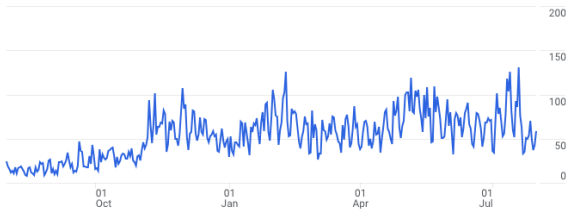


Figure 4: Growth of *sign.mt* users over the last year.

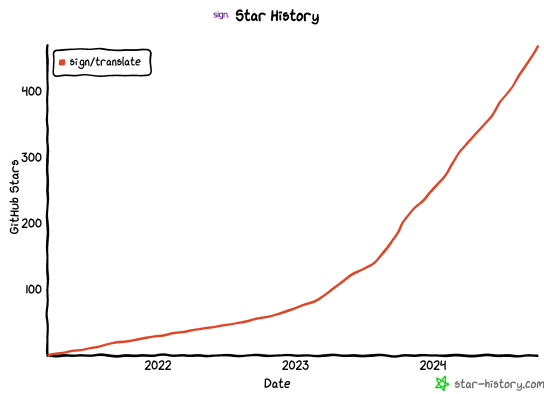


Figure 5: Number of stars for the repository over time.

Further validation of the community interest in *sign.mt* is evidenced by the increasing number of stars for its repository, reaching 470 stars as of October 6th, 2024 (Figure 5).

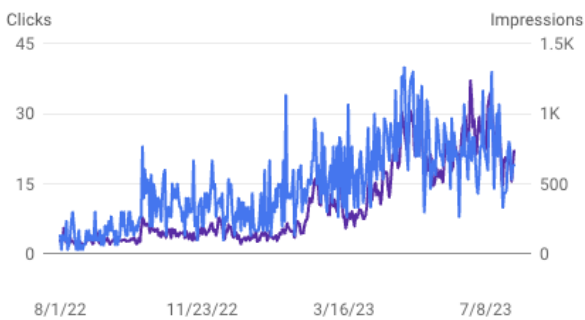


Figure 6: Google Search Console metrics showing increasing interest in *sign.mt*. (Clicks in blue)

Public interest in *sign.mt* is further supported by Google Search Console metrics (Figure 6), showing a significant increase in impressions and clicks over the past six months: 3.75K clicks (up from 1.56K), and 106K impressions (up from 24.4K). Despite the absence of a marketing team and a single maintainer, *sign.mt* has managed to carve a niche for itself in the realm of NLP tools, reiterating its significance and impact.

4 Distribution

The code for *sign.mt* is openly accessible and available for contribution on GitHub at <https://github.com/sign/translate>, under CC BY-NC-SA 4.0. Open sourcing with a permissive license encourages the continuous refinement and enhancement of the app through contributions from the wider developer and research communities.

The web application is freely accessible at <https://sign.mt>, designed with a responsive layout to cater to both desktop and mobile devices. Adhering to the design principles native to each platform, the application ensures an intuitive and user-friendly experience across all devices. With localization being a critical aspect of accessibility, the app interface supports 104 languages. Contributors can add their language or enhance the support for existing languages.

In addition to the web application, native builds for iOS and Android devices are also provided through the GitHub repository. While these are currently in development, the plan is to make them available on the respective app stores as they reach stability, thereby extending the reach of *sign.mt* to a wider audience.

Limitations

As an evolving open-source project, *sign.mt* still faces several challenges and limitations.

At present, the app does not provide complete support for every component of the translation pipeline. Notably, the SignWriting-to-pose animation model does not currently exist, and instead, we use a simple dictionary lookup approach (Moryossef et al., 2023b). Although it serves as an interim solution, it is insufficient for handling signed languages. We eagerly anticipate and encourage contributions from the research community to fill this gap with more advanced models.

Although the app aspires to be a multilingual platform, the availability of models for different languages is currently fragmented. We rely on the research community to develop and contribute models for different languages. The support for each language, therefore, depends on the respective models available, leading to varying degrees of effectiveness across languages. For example, the SignWriting translation module works reasonably well for English/American Sign Language, German/German Sign Language and Portuguese/Brazilian Sign Language translations, and

much worse for all other language pairs. Another example is the dictionary-based baseline only working on languages where dictionaries are available.

Due to the client-side deployment, we are restricted to using relatively smaller models. This inevitably leads to trade-offs in terms of translation accuracy and quality. While the offline functionality ensures accessibility in low connectivity areas, the constraint on model size is challenging.

The video processing components, including pose estimation and video rendering, are computationally intensive. This demands significant computational power, limiting the app's performance on devices with lesser computing capabilities. Optimizing these components further to ensure a smoother user experience across a wider range of devices is a challenge, often met with using lower-end models to achieve smoothness at the cost of accuracy.

Despite these limitations, *sign.mt* serves as a robust foundation upon which future advancements can be built. It continues to evolve in response to the feedback of the wider community, consistently striving towards the goal of facilitating accessible, inclusive communication.

References

- Rotem Shalev Arkushin, Amit Moryossef, and Ohad Fried. 2023. Ham2Pose: Animating sign language notation into pose sequences. pages 21046–21056.
- Nikolay Bogoychev, Jelmer Van der Linde, and Kenneth Heafield. 2021. TranslateLocally: Blazing-fast translation running on the local CPU. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 168–174, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ivan Grishchenko and Valentin Bazarevsky. 2020. *Mediapipe holistic*.
- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.
- Zifan Jiang, Amit Moryossef, Mathias Müller, and Sarah Ebling. 2023. Machine translation between spoken languages and signed languages represented in SignWriting. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1661–1679, Dubrovnik, Croatia. Association for Computational Linguistics.
- Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. *Mediapipe: A framework for perceiving and processing reality*. In *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*.
- Amit Moryossef. 2024. pose-anonymization: Remove identifying information from sign language poses. <https://github.com/sign-language-processing/pose-anonymization>.
- Amit Moryossef and Zifan Jiang. 2023. *Signbank+: Multilingual sign language translation dataset*.
- Amit Moryossef, Zifan Jiang, Mathias Müller, Sarah Ebling, and Yoav Goldberg. 2023a. Linguistically motivated sign language segmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.
- Amit Moryossef and Mathias Müller. 2021. pose-format: Library for viewing, augmenting, and handling .pose files. <https://github.com/sign-language-processing/pose>.
- Amit Moryossef, Mathias Müller, Anne Göhring, Zifan Jiang, Yoav Goldberg, and Sarah Ebling. 2023b. An open-source gloss-based baseline for spoken to signed language translation. In *2nd International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*. Available at: <https://arxiv.org/abs/2305.17714>.
- Mozilla Developer Network. 2020. *Intl.Segmenter*. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/Segmenter.
- OpenAI. 2022. *Chatgpt: Optimizing language models for dialogue*.
- Alex Salcianu, Andy Golding, Anton Bakalov, Chris Alberti, Daniel Andor, David Weiss, Emily Pitler, Greg Coppola, Jason Riesa, Kuzman Ganchev, Michael Ringgaard, Nan Hua, Ryan McDonald, Slav Petrov, Stefan Istrate, and Terry Koo. 2016. Compact language detector v3 (cld3). <https://github.com/google/cld3>. Accessed: 2023-08-01.
- Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883.
- Valerie Sutton. 1990. *Lessons in sign writing*. Sign-Writing.