UEH University

UEH Institue of Innovation

FINAL REPORT

Sign Language to Text Translation System Using Machine Learning

Student team:	Team 09
Student name & ID:	Tạ Khánh Hà - 31221023776
	Nguyễn Xuân Hân - 31221023021
	Nguyễn Phương Nghi - 31221020433
	Võ Trần Bảo Trân - 31221024798
	Nguyễn Thị Cẩm Tú - 31221021819
Academic supervisor:	Ph.D Nguyễn Thiên Bảo

Ho Chi Minh, November 05, 2024

Contents

Abstract

The main goal of this project is to create a tool that can translate sign language into languages to help people who don't know sign language communicate with the community more easily. Currently the project is concentrating on English, German and French. The website is set up as a user translation platform, like Google Translate that allows for translations in both directions. This project uses machine learning and computer vision technologies to identify and translate sign language gestures into text and vice versa making communication smoother, for everyone involved. We strive not to offer a means, for translating sign language but also to foster inclusivity and equality, in sharing information among all individuals.

1. Introduction

1.1. Project Objectives

The objective of this project is to develop a multilingual sign language translation tool, similar to Google Translate. Its main supported languages will include English, German, and French. Users will be able to input text on which the tool can respond with a video in sign language, hence helping non-signers and the deaf interact better. It will also facilitate translation from sign language to text via video uploads, hence making the platform more accessible for the deaf to present their messages.

1.2. Rationale for Selecting the Topic

Advancements in automated translation technology have made it easier for millions of people to break down language barriers and interact using tools like Google Translate. However, a crucial aspect that hasn't received focus in today translation tools is sign language. This unique form of communication plays a role for the community in engaging with society and sharing their feelings and ideas. The lack of sign language translation tools has posed challenges for individuals with hearing impairments when communicating with those who're not familiar with sign language. Many people who want to learn and comprehend sign language encounter difficulties because of the availability of convenient learning materials. This emphasizes the importance of having a tool that can translate sign language to written language and vice versa.

The project aims to provide numerous benefits and contributions to society; it will serve as a crucial tool for communication between the deaf and the broader community, reducing reliance on interpreters and fostering an inclusive communication environment. Furthermore, the tool will offer significant value for education and research, enabling individuals to easily access and learn sign language. Ultimately, the project will contribute to the establishment of a society free from language barriers, where sign language is widely recognized, thereby promoting linguistic diversity and equality in communication.

2. Literature Review and Related Works

Sign language translation (SLT) facilitates communication between deaf and hearing individuals by converting sign language into spoken or written language. There are two primary approaches in SLT: gloss-based and gloss-free methods. Gloss-based methods use intermediate representations called glosses-textual annotations of sign language movements-while gloss-free methods aim for direct translation from sign language videos to text without intermediate annotations. This review discusses prominent models in both approaches, highlighting their methodologies, strengths, weaknesses, and citing relevant research papers.

2.1. Gloss-Based Sign Language Translation Models

Model name	Description	Reference
1. Neural Sign Language Translation	Camgoz et al., 2018 introduced a model using CNNs and Bi-LSTM networks for Sign2Gloss and an attention-based Seq2Seq model for Gloss2Text. Strengths include structured learning and improved accuracy, while weaknesses involve annotation dependence and error propagation.	Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2018). Neural Sign Language Translation. CVPR, 7784-7793. [4]

2. Sign Language Transformer	Yin et al., 2020 introduced a transformer-based model leveraging gloss annotations. It includes Transformer Architecture (self-attention for long-range dependencies) and Multi-Task Learning (joint training for recognition and translation). Strengths are enhanced contextual understanding and joint optimization. Weaknesses include high computational and data demands.	Yin, S., Xia, Z., Chen, X., Zhou, H., & He, S. (2020). Sign Language Translation with Transformer. MM '20, 1778–1786. [5]
3. Neural Sign Language Translation by Learning Tokenization	Orbay & Akarun, 2020 introduced a model that learns sub-word tokenization of glosses, with Tokenization Mechanism and End-to-End Training for efficiency. Strengths are vocabulary efficiency and flexibility. Weaknesses include added complexity and dependence on gloss annotation quality.	Orbay, E., & Akarun, L. (2020). Neural Sign Language Translation by Learning Tokenization. arXiv:2004.03519. [6]

2.2. Gloss-Free Sign Language Translation Models

Model name	**Description**	**Reference**
1. Sign Language Transformers	Camgoz et al., 2020 developed a gloss-free transformer model with End-to-End Architecture (no gloss representation) and Spatial-Temporal Encoder (CNNs and transformers for video input) plus a Transformer Decoder for target text. Strengths are no gloss annotations needed and nuanced capture of sign input. Weaknesses are data intensiveness and training complexity.	Camgoz, N. C., Hadfield, S., Koller, O., & Bowden, R. (2020). Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. CVPR, 10023-10033. [7]
2. Progressive Transformers	Saunders et al., 2020 proposed a progressive transformer model with Progressive Learning (intermediate supervision) and Multi-Cue Integration (manual and non-manual cues). Strengths include enhanced feature learning and rich contextual information. Weaknesses are increased model complexity and high resource demand.	Saunders, B., Camgoz, N. C., & Bowden, R. (2020). Progressive Transformers for End-to-End Sign Language Production. ECCV, 687–705. [8]
3. Self-Supervised Learning for SLT	Pu et al., 2019 explored self-supervised learning for gloss-free SLT using Cross-Modal Training (pre-training on related tasks) and Feature Extraction without labeled data. Strengths are data efficiency and improved generalization. Weaknesses include limited improvement without large datasets and additional design complexity.	Pu, J., Zhou, P., Wang, F., & Xu, W. (2019). Boosting Continuous Sign Language Recognition via Cross Modality Augmentation. CVPR, 11509-11518. [9]

Comparison and Discussion

- Data Requirements: Gloss-based models can perform well with smaller datasets if gloss annotations are available. Gloss-free models require larger datasets but eliminate the need for costly annotations.
- Model Complexity: Gloss-free models are generally more complex due to their end-to-end nature and need for extensive feature extraction.
- Performance: While gloss-based models currently have an edge in performance due to structured intermediate representations, gloss-free models are rapidly improving with advances in deep learning techniques.
- Flexibility and Scalability: Gloss-free models are more adaptable to different sign languages and can scale better in multilingual contexts.

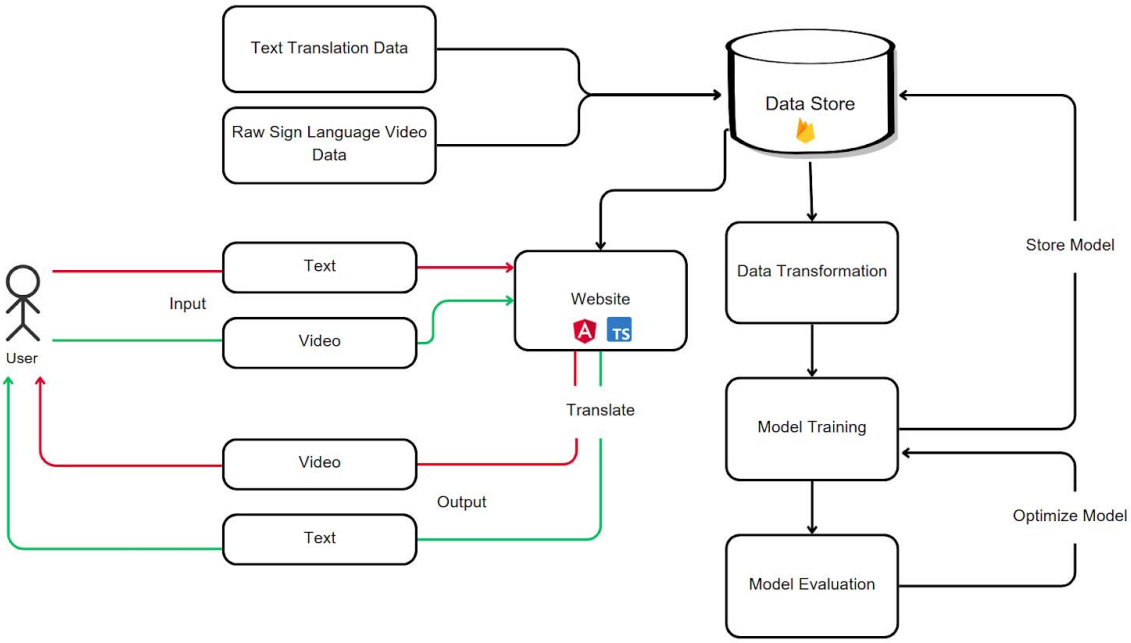
Conclusion

Both gloss-based and gloss-free models have made significant contributions to the field of sign language translation. Gloss-based models offer high accuracy when gloss annotations are available but are limited by data scarcity and potential loss of nuanced information. Gloss-free models present a scalable alternative that can leverage larger datasets without annotations, though they require more data and computational resources. Future research may focus on hybrid models that combine the strengths of both approaches or explore advanced self-supervised learning techniques to enhance gloss-free models.

3. Methology

The team believed it was important to conduct a deep study of the architectural system for the sign language translation tool, given that existing models and products had many limitations. The architecture designed is described in detail in Section 3.1, with the objective of optimizing performance toward improving the translation and user experience in raising conditions of communication to a higher level in accord with the needs of both the deaf and those who do not know sign language.

3.1. System Architecture Overview



A user will input text or video in sign language and after processing the data, this system will return results through the website interface. There are two major sources of data: the text translation data, and the raw data in the form of sign language videos. These are to be stored in a Data Store for maintenance and management of all data, including the versions of the different models trained. This stage of Data Transformation will convert raw data into a format suitable for training models, and this ensures that translation models receive uniform standardized data. These models will later be trained in the Model Training stage for developing the capability to translate sign languages from texts and videos accurately. Afterwards, the models undergo Model Evaluation & Optimization to perform better with higher accuracy. Optimized models shall be stored at the Data Store, ready to be used in translation processes. Web interface serves as a bridge between users and the system, whereby users input their request and obtain translation results in either text or sign language video format.

3.2. AI model Architecture Overview

- **Approach 1**

Our first approach was to improve on the Stanford students' research, which made use of the MS-ASL dataset and the I3D model. The objective was to apply a Sign2Gloss2Text technique in order to improve this model. But while developing, we found 2 problems. First, we came into a few problems when we first downloaded the MS-ASL dataset. The vocabulary data was provided in the JSON files that contained the train, test, and validation sets. With start and end times indicating the section of the video for the intended sign, each word entry matched a link to a YouTube video. However, we had several

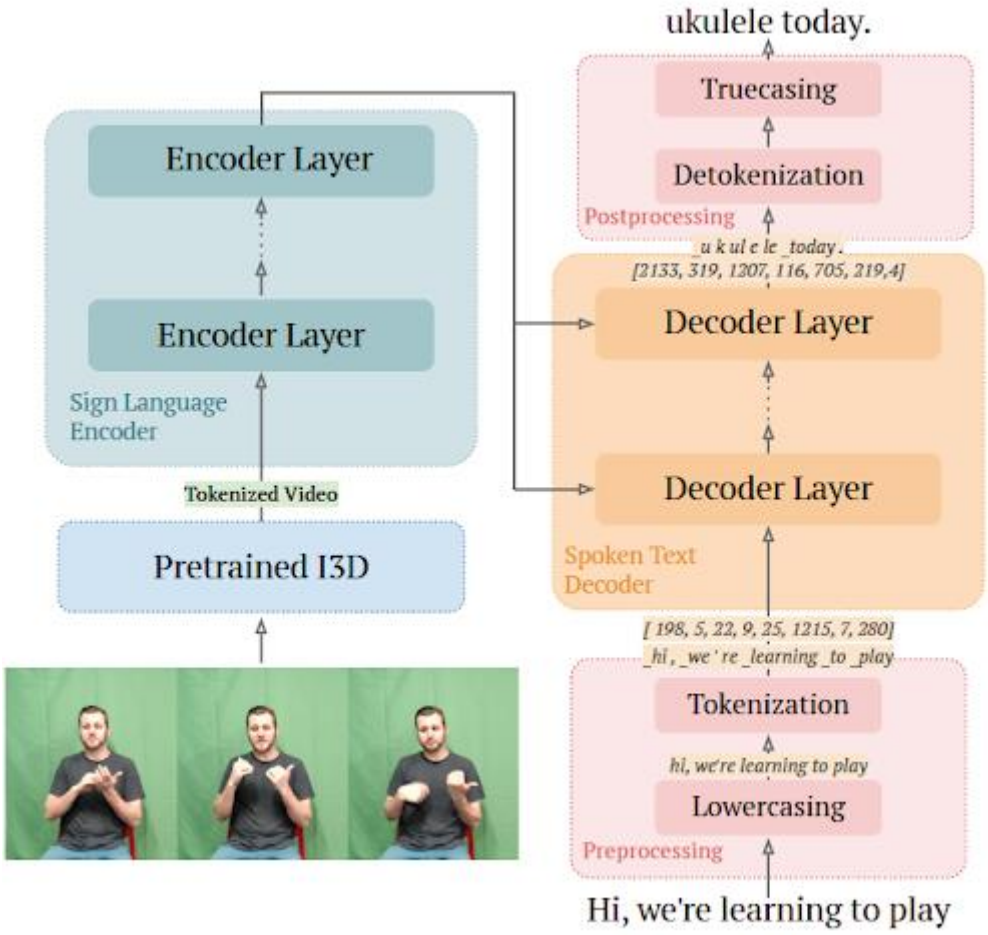
difficulties when downloading. Despite the training set JSON including more than 16,000 words, only roughly 1,600 words were downloaded successfully. After examining the dataset, we found that a large number of the YouTube links were either private or no longer accessible. Furthermore, it was often difficult to access parts because the download script pulled the full video rather than just the relevant sections.

Simultaneously addressing the dataset challenges, our team commenced the development of the Sign2Gloss2Text model. Although we had no serious problems during the Sign2Gloss phase, we quickly discovered that the Gloss2Text stage was quite difficult. The model needs a dataset with gloss-to-text mappings for this component to function, yet MS-ASL does not offer this dataset. This delays us from completing the Sign2Gloss2Text pipeline as intended.

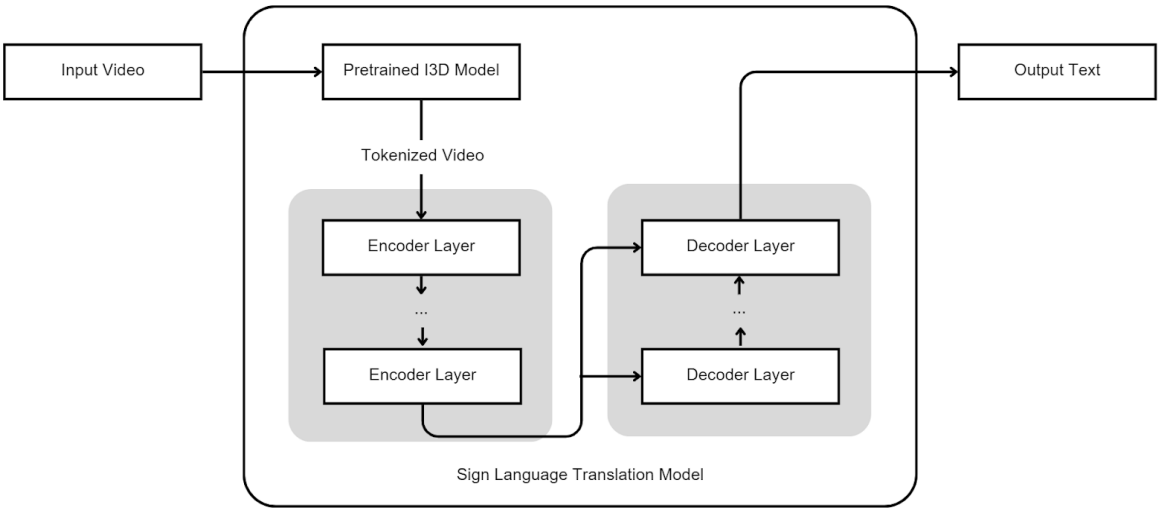
To solve this problem, we adopted a second approach, using a Transformer encoder-decoder model with the How2Sign dataset.

- **Approach 2**

In our second method, we translated sign language to spoken language (Sign2Text) directly without the need for a gloss stage using a simple Transformer encoder-decoder paradigm. This model uses a transformer architecture trained on video features retrieved by a pretrained I3D network, as explained in Sign Language Translation from Instructional Videos (CVPR 2023). In order to produce appropriate English text, the transformer processes the spatial and temporal characteristics that the I3D model extracts from continuous sign language videos.



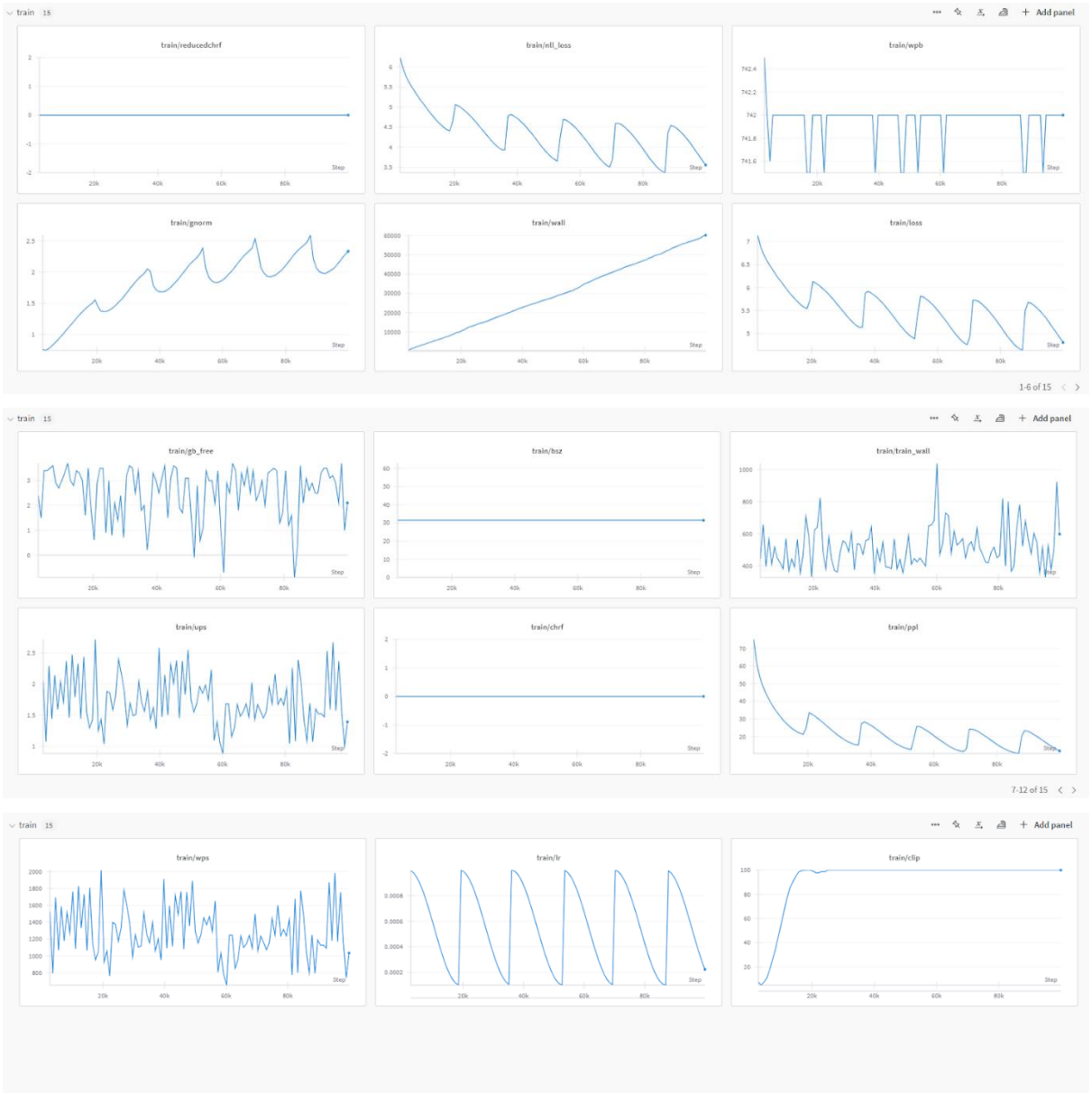
The architecture of this model operates by first passing video data through a pretrained I3D model to extract features, with the output saved as numpy files. These files are then processed by two Transformer layers: an encoder and a decoder, which generate the final translation.



The advantage of this model design is that it improves contextual awareness of the generated sentences by concentrating on the most important portions of the video sequence during translation. Our findings, however, point to two important drawbacks with this paradigm. First, despite improvements to the model and dataset, translation results were not as encouraging as expected due to low BLEU scores. The BLEU score plateaued at about 8 after multiple training epochs, which is insufficient for useful, real-world applications. The difficulty of comprehension of sign language and the model's inability to comprehend lengthy or complicated words account for this. The duration of the training time is the second major barrier. With a single NVIDIA GeForce RTX 3050 Ti GPU, our configuration takes about 18 hours to run 108 epochs, which slows down the iteration process for testing and improving model variations. This restriction makes it more difficult for us to effectively improve the model and test various configurations.

- Results on val dataset:

- Some train results:



- **Approach 3: Sign2Signwriting2Text và Text2Signwriting2Sign**

We have taken a different approach as a result of these constraints, and this is the strategy we decided to use for our project: employing SignWriting as a translation intermediate. SignWriting, as instead of gloss, is a writing system designed especially for sign languages. More accurate translation is made possible by our method's use of pretrained models [3]. This method's ability to achieve a higher BLEU score than the prior model is one of its main advantages. Section 4 provides more information on this strategy.

4. Experiment

4.1. System

The website system for translating sign language into text and vice versa, developed by the team, consists of two main components: the frontend and the backend.

- **Frontend**

The frontend is built with Typescript and the Angular framework to ensure a user-friendly and intuitive interface.

- **Backend**

The backend is developed in Typescript with the Firebase framework, handles video processing, from extracting body features and frames to applying the AI model for sign language translation.

- **Database**

The system's database is deployed through Firebase Firestore, a NOSQL database suitable for real -time applications, allowing storage of translation information, user history, and versions Record the identified gestures. Firestore operates according to the document model and collection, allowing data storage in a structured and flexible manner.

The data table structure includes the following tables:

- **Users:** save information about user such as name, email address, phone.
- **Translations**: Save the translations of the user, the epidemic and the time of saving.
- *Table 1: Users*

Field	**Data Type**	**Constraints**	**Description**
uID	INT (PK)	AUTO_INCREMENT	Unique ID for each user
translationID	INT (FK)	AUTO_INCREMENT	Unique ID for each translation
displayName	VARCHAR(255)		User’s name display on screen
firstName	VARCHAR(255)		User’s first name
lastName	VARCHAR(255)		User’s last name
email	VARCHAR		User’s email
phone	INT		User’s number phone
address	VARCHAR(255)		User’s address

- *Table 2: Translations*

Field	**Data Type**	**Constraints**	**Description**
translationID	INT (PK)	AUTO_INCREMENT	Reference to the Users table
translationType	VARCHAR(255)	NOT NULL	Type of translation: “spoken to sign” or “sign to spoken”
text	VARCHAR(255)	NOT NULL	Translated text or imported text
signLanguage	VARCHAR(255)	NOT NULL	Sign language input or sign language output “Video.mp4”

saveAt	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time when user saves the translation

4.2. AI Architecture

4.2.1. Introduction to Signwriting

As a middle stage, to create the sign language translation (SLT) task using a sign language writing system (shown in Figure 1): We suggest converting spoken language text into written sign language and then turning this intermediate output into a final video or posture output and in reverse. In this work, we examine the translation between spoken and written signed languages using this multi-step approach of SLT. The intermediate writing system is called SignWriting.

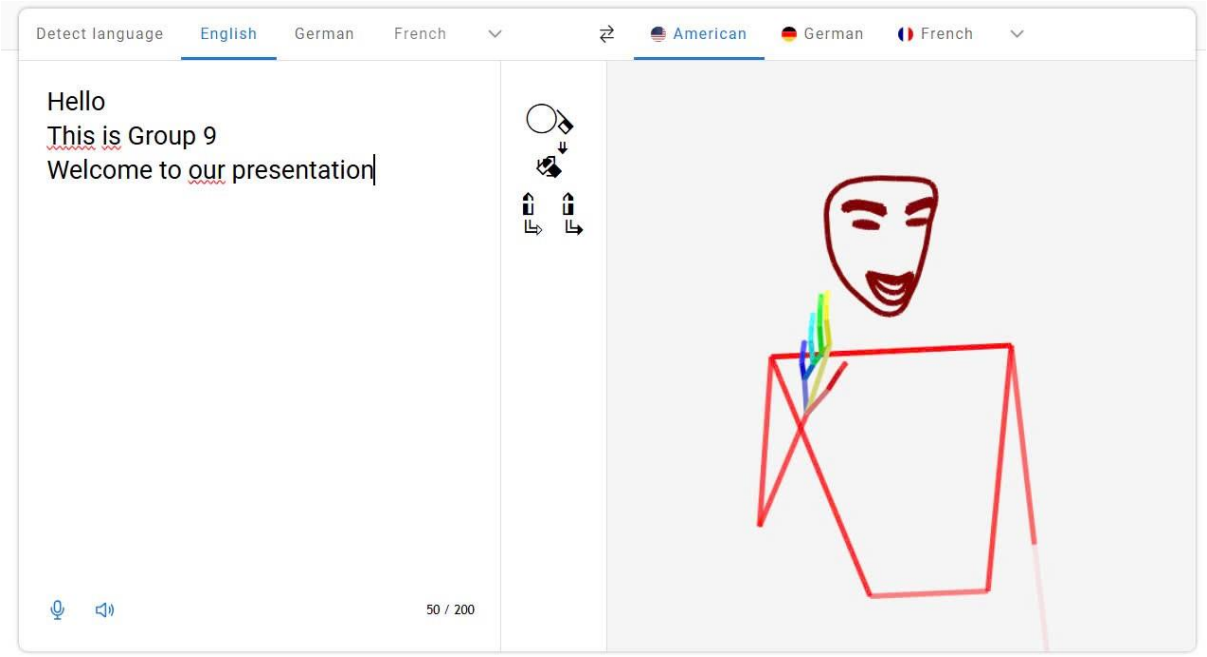


Figure 1: Demo application based on our models, translating from spoken languages to signed languages represented in SignWriting, then to human poses.

- **Sign Writing**

SignWriting (Sutton, 1990) is a featural and visually iconic sign language writing system. Previous work explored recognition (Stiehl et al., 2015) and animation (Bouzid and Jemni, 2013) of SignWriting.

SignWriting has many advantages, including being computer assisted, universal (multilingual), very simple to learn, and extensively documented. Furthermore, even though it appears pictographic, the writing system is clearly defined. Each sign's location on a two-dimensional plane and a series of symbols (box markers, graphemes, and punctuation marks) can be written down.

SignWriting has two computerized specifications, Formal SignWriting in ASCII (FSW) and SignWriting in Unicode (SWU). SignWriting is two-dimensional, but FSW and SWU are written linearly, similar to spoken languages. Figure 5 gives an example of the relationship between SignWriting, FSW, and SWU. In light of the article we cited, we also concentrate on FSW. [1]

Figure 2: Hand shapes and their equivalents in SignWriting. [1]

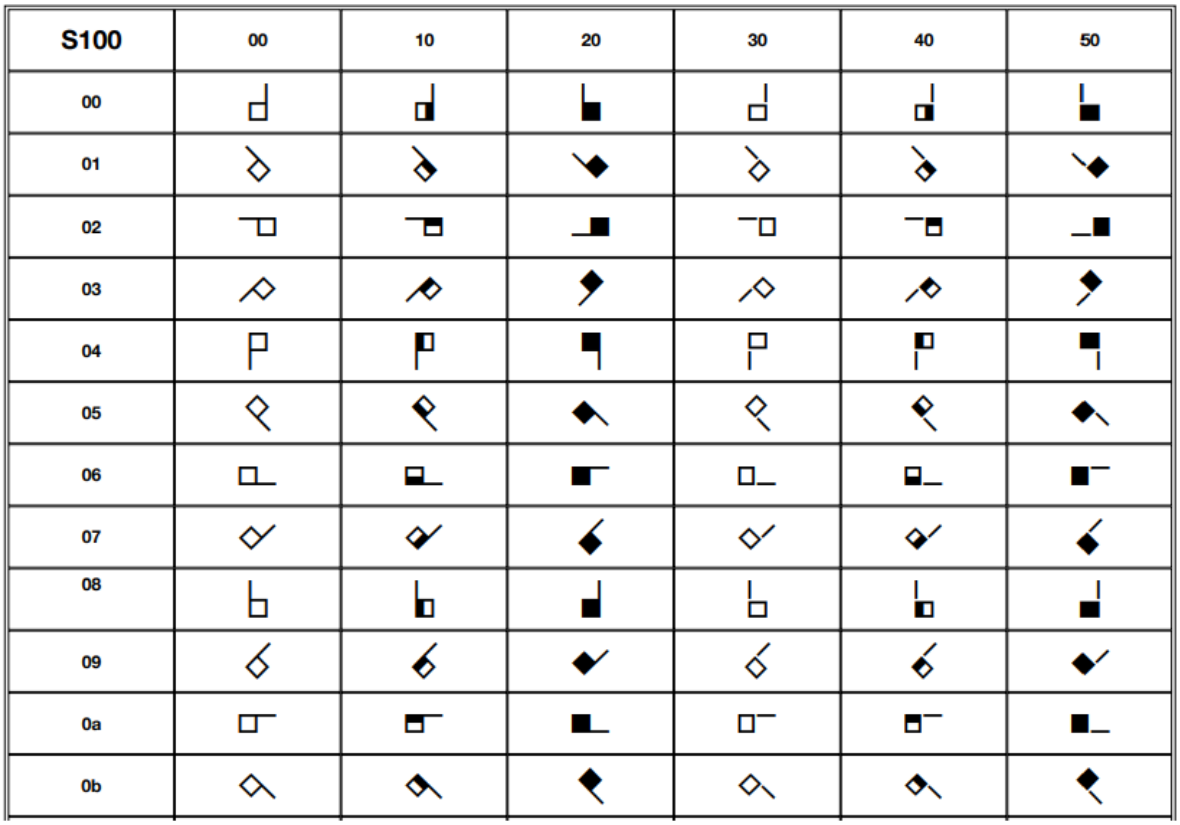


Figure 3: Orientation of a symbol in SignWriting in 3D space. Each row applies a rotation of the palm in a 2D space vertical to the ground. Each column applies a rotation of the palm in a 2D space parallel to the ground. This can be seen as a factorization of the symbol S100xx to its core S100 plus row and column numbers. [1]

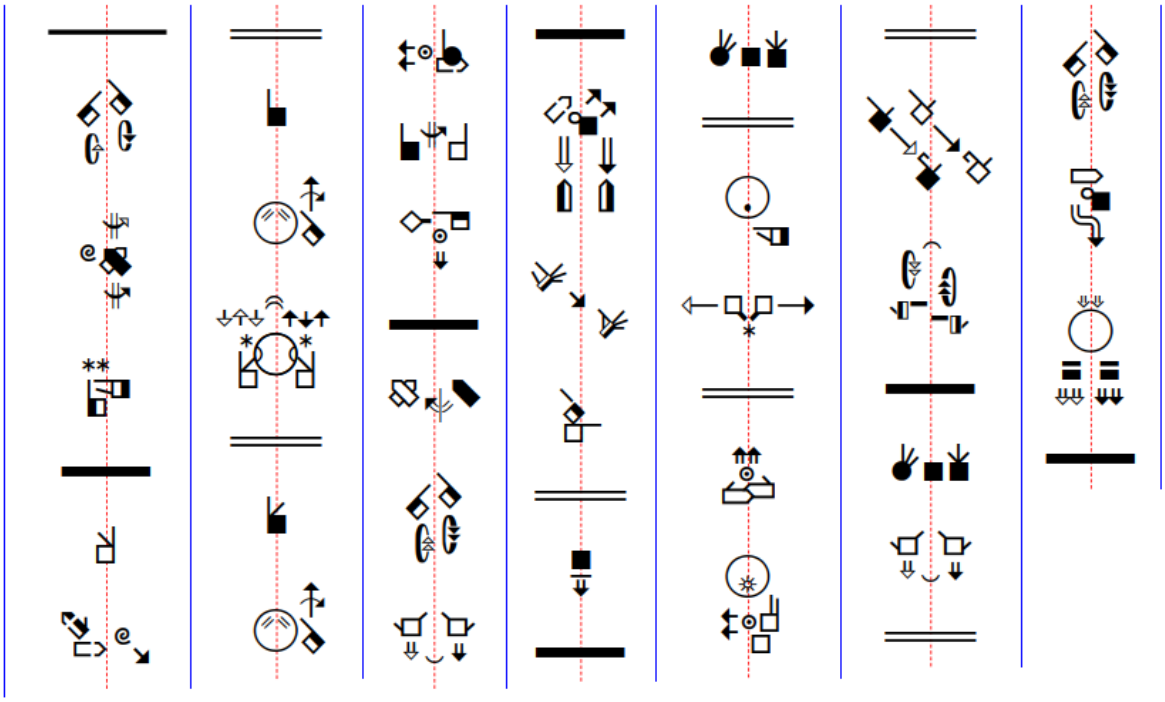


Figure 4: An example of SignWriting written in columns, ASL translation of an introduction to Formal SignWriting in ASCII. The relative positions of the symbols within the box iconically represent the locations of the hands and other parts of the body involved in the sign being represented. [1]

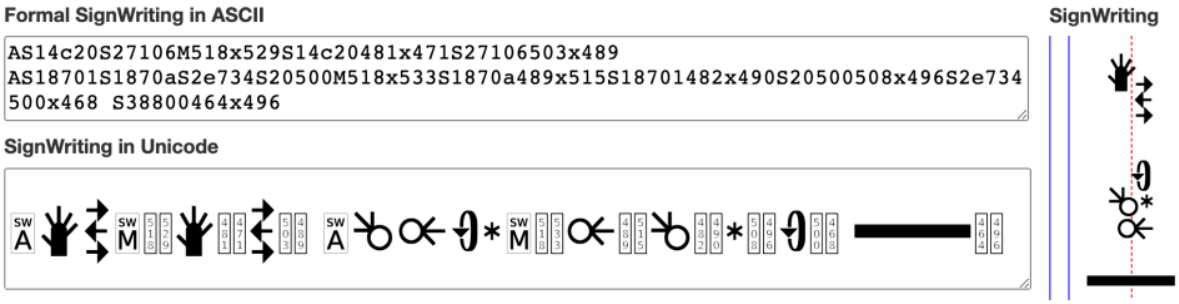


Figure 5: "Hello world." in FSW, SWU and SignWriting graphics. In FSW/SWU, A/SWA and M/SWM are the box markers (acting as sign boundaries); S14c20 and S27106 (graphemes in SWU) are the symbols; 518 and 529 are the x, y positional numbers on a 2-dimensional plane that denote symbols' position within a sign box, S38800 (horizontal bold line in SWU) is the punctuation full stop symbol. [1]

We use the capabilities of Amit Moryossef's pretrained model [2] to produce hand shapes and facial features that enhance the visual components of SignWriting in order to generate SignWriting symbols. By identifying and replicating these particular characteristics, our methodology generates structured outputs in a SignWriting-compatible way.

To identify and encode important hand and facial traits, we would enter video or position data, which the model would then process. It is therefore possible to transfer these encoded symbols into FSW or SWU format SignWriting representations. We can automate a large portion of the symbol creation process by using this model as a foundation, freeing up time to improve the arrangement, placement, and relative orientation of symbols to guarantee a precise and readable SignWriting output.

4.2.2. Details on how to build the model

- **Data**

The data source used for model pre-trained is SignBank.

- **Model**

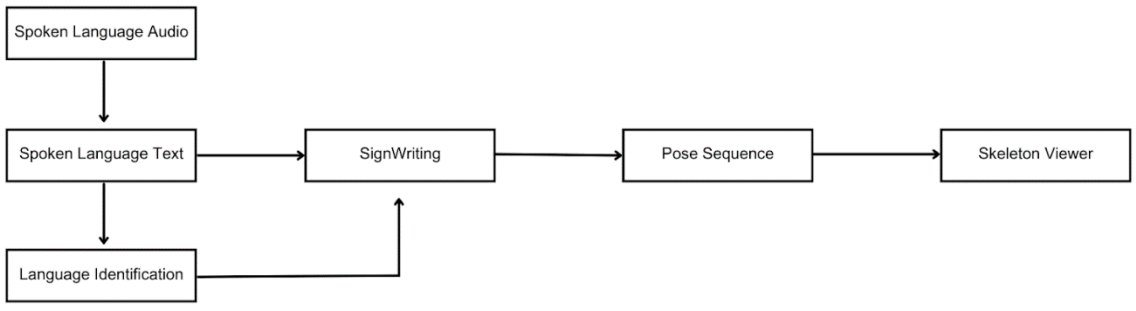


Figure 6: Sign language text into Skeleton Viewer

This model illustrates the way spoken language text and audio can be translated into sign language expression. Text and audio inputs in spoken language are used first. In order to ensure accurate processing, Language Identification assists in identifying the language of the input text or voice. After the spoken language text has been discovered, it is converted into SignWriting, a written method that graphically depicts sign language. A Pose Sequence, which specifies the precise hand

and body motions needed to express each sign, is created from this SignWriting output. A Skeleton Viewer, a program that shows a virtual skeleton performing the translated sign language, is then used to visualize these pose sequences.

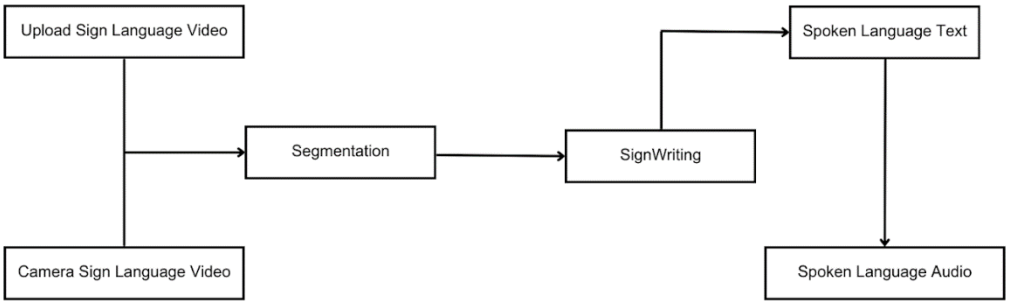


Figure 7: Sign language videos into spoken language text and audio output

By defining the process for converting videos in sign language into spoken language text and audio output, this model allows communication between sign language users and non-signers. There are two choices for entering sign language v ideo data at the start of the procedure. Users have the option to stream live using a webcam that records real-time signing or upload a pre-recorded video in sign language. In the following step, known as segmentation, the model examines the video to separate and recognize distinct signs or sentences. This segmentation stage is essential because it allows the system to separate specific sign language elements from the video frames, readying them for precise translation.

These sign components are converted into SignWriting symbols after they have been segmented. SignWriting is a visual writing system created especially for sign languages that includes facial expressions, hand shapes, and gestures. This model converts visual data into a structured symbolic form by using Amit Moryossef's pretrained model to assist in generating the hand forms and facial features in the SignWriting format. A language model that recognizes and interprets SignWriting as text makes it easier to translate these symbols into spoken language text.

This entire process improves communication accessibility across language modes by ensuring that sign language inputs, whether from live broadcasts or video uploads, are available in spoken language form through both text and audio.

5. Results and Analysis

5.1. Introducing Metrics

In the field of machine translation, the BLEU-4 and ChRF measures play an important role in assessing the accuracy of translation models, including the models from text to symbol language (text2sign) and from the symbol language to the text (sign2text).

- **BLEU-4**

BLEU (Bilingual Evaluation Understudy) is one of the most common measures to evaluate machine translation models. The BLEU-4 version, focusing on 4-grams, measuring the accuracy of the phrases consisting of 4 consecutive words between the translation of the model and the reference data. Specifically, BLEU-4 compares the matching of the 4 words phrases, reflecting the accuracy of context and meaning in the translation. BLEU-4 score can be represented from 0 to 1 or from 0 to 100 if calculated by the percentage; High score shows good matching between model and original translation.

In the application of the BLEU-4 for Text2Sign, this measure helps assess the ability of the model to accurately convert important phrases from text to symbols, while maintaining the core meaning. For sign2text, BLEU-4 assesses the accuracy when the symbol language conversion model returns into text, especially in key phrases.

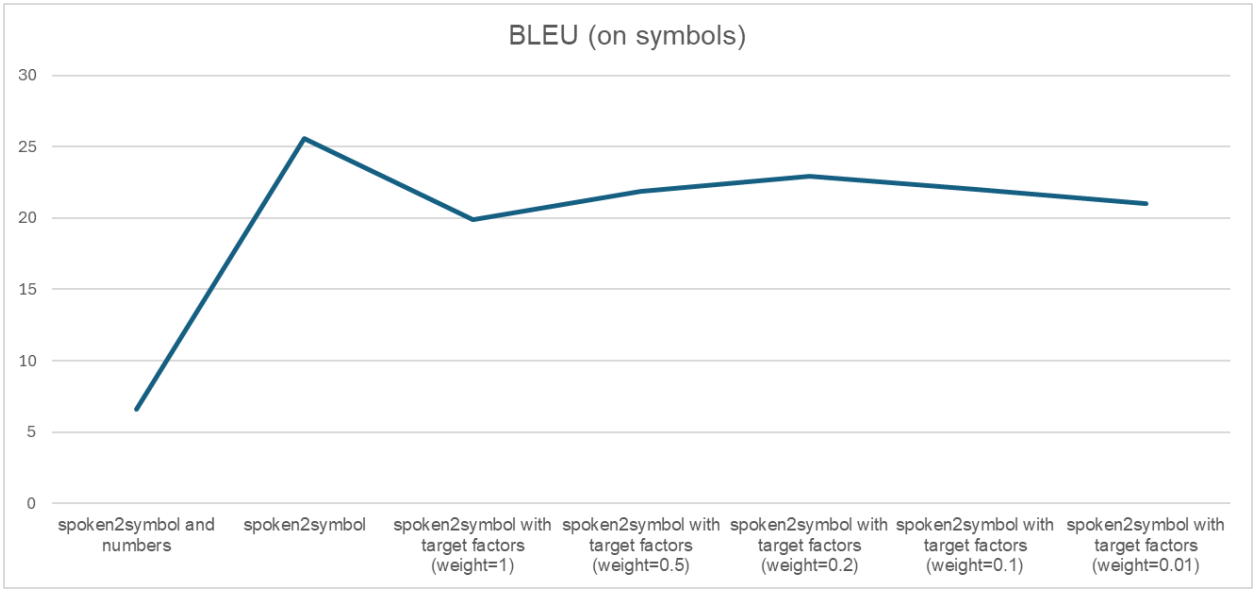
- **ChrF**

ChRF (Character F-Score) is a measure based on characters, developed to meet the requirements of evaluating languages with complex structures or containing shortened elements, such as symbol language. The ChRF is based on the ChRF2 ++, the advanced version of the ChRF, evaluating the accuracy and coverage at the character level, bringing accuracy to the structure of the translation. The ChRF measure is especially useful for the sign language because it has the ability to capture small differences in structure, expressed through each specific symbol or sign.

In the application of ChRF for Text2Sign, this measure helps determine the similarity between the output of the model and the original, thereby supporting the accuracy assessment of each specific symbol. For sign2text, the ChRF measures the model's ability to accurately decode each symbol, ensuring that the semantics of the sign language are fully and accurate.

5.2. Results

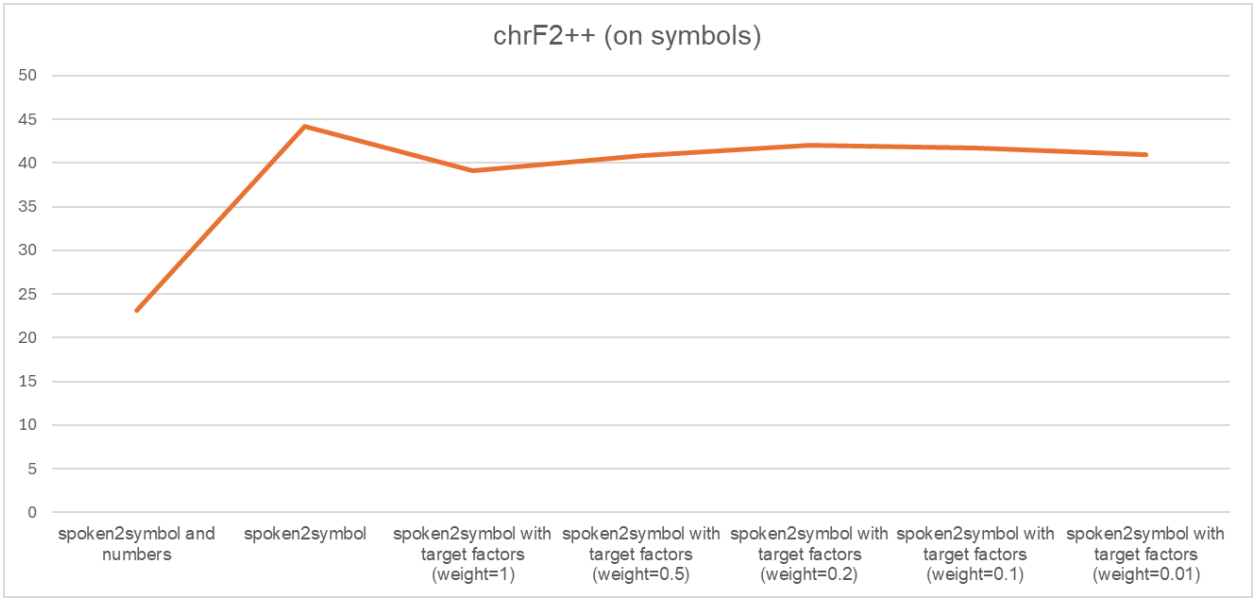
5.2.1. Text2Sign



Based on the BLEU result table for models that convert from spoken language into signs (text2sign), we can draw some important comments on the overall performance of the models.

The “Spoken2Symbit” model achieves the highest BLEU point of 25.6, showing that it works effectively in the conversion from the text to the symbol without additional elements. In contrast, the “Spoken2Symbol and Numbers” model only reaches BLEU as 6.6, which proves that adding digital components has significantly reduced the accuracy of the model. The reason may be due to the complexity of this factor that makes it difficult for the model to accurately capture the symbols.

When considering the effects of the weight of additional factors, we see that the BLEU performance varies depending on the applicable weight. Specifically, with the number 1, the BLEU score reached 19.9, lower than the model only “Spoken2Symbol” (25.6), showing that the use of additional factors with maximum weight does not bring high efficiency. When the weight decreased to 0.5, the BLEU point increased to 21.9, and reached the highest when the weight was 0.2, with the BLEU point of 22.9. This shows that the use of additional factors with average weight can improve accuracy, but should not exceed certain weight. When the weight continues to decrease to 0.1, the BLEU point remains at 22, but when it drops to 0.01, the BLEU point decreases to 21. This indicates that decreasing weight to too low can reduce the effectiveness of additional factors.

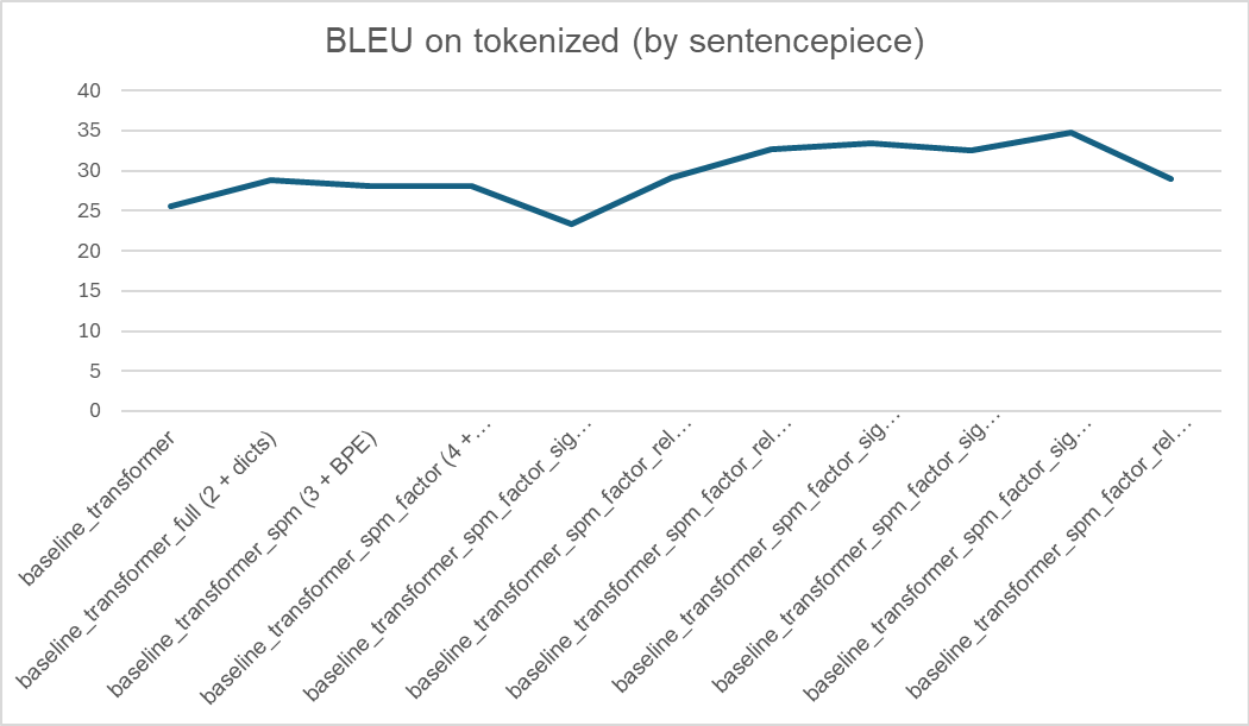


The overall efficiency of the models shows that the "Spoken2Symbol" model achieves the highest chrf2 ++ score of 44.2, showing the best performance in switching from text to symbols without additional elements. This result is consistent with the previous BLEU index, proving that the basic model operates effectively without further adjustment. In contrast, the "Spoken2Symbol and Numbers" model achieves the lowest Chrf2 ++ score of 23.1, showing significantly less performance when there is an additional digital component, possibly due to the increased complexity from this factor.

When analyzing the effects of the weight of additional elements, we realize that the chrf2 ++ score changes the same as the BLEU index, but it seems more stable. With the number 1, the score of ChRF2 ++ is 39.1, lower than the pure "Spoken2Symbol" model (44.2), showing that the maximum use of the additional weight is not the optimal choice. When the weight decreased to 0.5, the ChRF2 ++ point increased to 40.8, and continued to peak at 0.2 with a point 42, showing that the average weight helps improve performance. Even if the weight is reduced to 0.1 and 0.01, the ChRF2 ++ score is still relatively high, respectively 41.7 and 40.9, respectively, showing that the weight reduction does not affect too much on this index while it may reduce noise.

When comparing between BLEU and ChRF2 ++ indicators, both showed the "Spoken2Symbit" model that works best without additional factors. The average weight (from 0.1 to 0.5) gives additional elements to help the model achieve better results than other weights (1) or too low (0.01).

5.2.2. Sign2Text



The overall performance of the models shows the basic model “Baseline_transformer” achieved BLEU 25.6, which is the starting point to compare with the optimized models. Adjustable models often have higher BLEU points, showing the effectiveness of refining in improving the accuracy of the model.

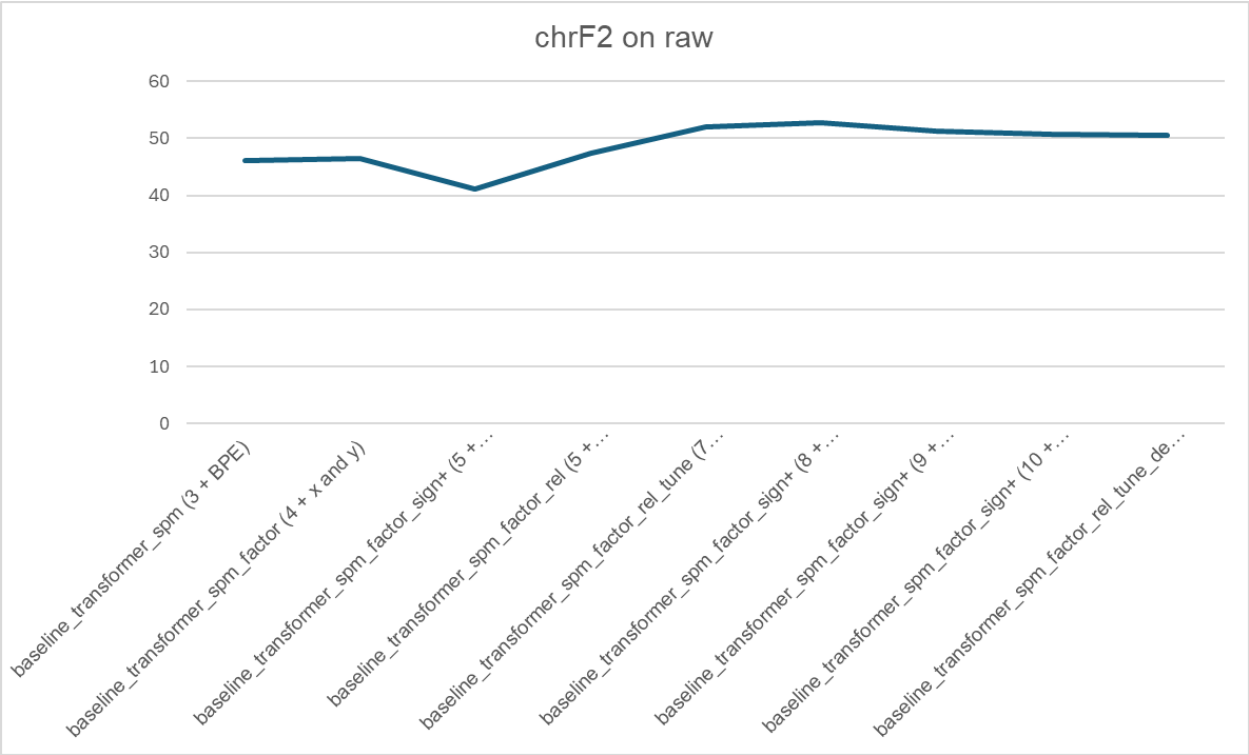
When considering the effects of various technical improvements, the “full” (2 + dicts) configuration with the full dictionary helps to increase the BLEU to 28.8, showing the use of the dictionary is beneficial for the results pandemic. Models “Baseline_Transformer_Spm” and “Baseline_Transformer_Spm_Factor (4 + X and Y)” Achieving BLEU points are 28.1 and 29.1, showing that the tokenization with SentencePiece and the addition of additional elements for X and Y coordinates can be Help the model better grasp complex symbols.

However, when adding FSW (Factored Signwriting) as an additional element in the “Factor with FSW (5 + Factored FSW Graphemes)”, the BLEU point is reduced to 23.4, this may be due to this supplement to this factor. Increase complexity without improving model efficiency. Meanwhile, the use of relative X and Y elements in “Factor with relative X and Y (5 + Relative x and Y)” helps to achieve BLEU 29.1, higher than absolute factors, shows that the relative factor can help the model better understand the context.

The adjustment of Dropout with “tieD_softmax” in the model (7) has helped to increase the BLEU to 32.7, showing that the application of a reasonable Regularization improves accuracy. Using FSW as an additional element in the model (8) to achieve the BLEU 33.5 point, higher than using both FSW for both source and additional factors, showing that only FSW is used for additional factors is how to be how to More optimal. When not using Lowercasing in the model (9), the BLEU point reaches 32.6, showing that keeping the capital letters can increase the accuracy of some specific words in the context.

Reducing BPE Vocabulary size from 2000 to 1000 of the model (10) has led to the BLEU point of 34.8, which is one of the highest results, showing that reducing the size of the BPE vocabulary can help the model be more optimal, it may be due to noise from unnecessary tokens. Although adding “de dict” in the model (10) only achieved the BLEU 29 point, showing a slight decrease, this shows that this dictionary may not be optimal in this case.

Conclusion, models that have adjusted Dropout, tieD_softmax, and FSW element when used appropriately have significantly improved performance compared to the basic model. Configuration “Baseline_transformer_spm_Factor_sign+ (10)” with a smaller vocabulary size of BLEU 34.8, is the highest result, showing that the model can be optimized with a smaller vocabulary. Factors such as keeping capital letters and using relative coordinates for symbols are small but useful improvements. The best model for the translation of sign2text in this case is “baseline_transformer_spm_factor_sign+ (10)” with a smaller vocabulary, because it reaches the highest BLEU point.



The basic model (Baseline_Transformer) reaches the lowest ChRF2 point, showing that adding additional processing techniques can improve the performance of the model. The addition of BPE (Baseline_Transformer_SPM) has improved its performance, proving that coding-code can enhance the capability of the model's context. Using factors to model graphemes (baseline_transformer_spm_factor) often brings better performance, this shows that exploiting information about graphemes can help the translation model more accurately. The method of combining elements (sign+ and reling) often achieved better results than using each individual method, this proves that combining information from many sources can improve the efficiency of tissue effectiveness. image. Finally, improving the Fine-Tune (reliable) can raise the ChRF2 score, showing that the model optimization for specific data sets is more effective.

6. Conclusion and Future Work

6.1. Conclusion

The current project has achieved significant results in developing a sign language translation system. It enables the conversion of text into sign language and vice versa from sign language to text for three countries: America, France, and Germany. Besides, the project is extended for more sign languages, though the extension is incomplete.

The project has been able to demonstrate that it is very possible to come up with a communication support tool for the deaf within the community. This would not only avail users with easy access to information but also promote their capabilities of communicating effectively with persons who do not use sign language. Also, the project has extended the ability to translate not just American Sign Language but some other countries' sign languages, thus improving communication for the hearing impaired in various nations.

Another limitation of this project in translating spoken language to sign language is that sign language grammar has not been perfectly portrayed yet. Grammatical rules of the signs differ from the ones used in spoken language, and the translation system has not yet been perfected to reflect these structures in results of translations. Because of this, some of the translations could be tricky to understand and sometimes look unnatural.

6.2. Future Work

In the future, one of the main aims is to complete the system, as there are still limitations to address. Additionally, improving the service by adding more supported languages is a key goal. The system will be further updated to support more languages and provide an accuracy that is at par with the three currently supported languages. This will enhance not only globalization but also user satisfaction across different countries.

Other key enhancements shall be put in place with a focus on overcoming what can be termed the worst limitation of the existing system is the grammatical accuracy that has failed to fully translate spoken languages into sign language. Therefore, the future enhancement of the system shall be focused on developing a more accurate translation model, which shall adhere to the grammatical rules of sign language. The translated text shall always provide proper and natural meanings for the hearing-impaired person, making communication easier.

References:

[1] 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 1706–1724, Dubrovnik, Croatia. Association for Computational Linguistics.

[2] Code on Google Colab. Amit Moryossef. SignWriting Hands Classification v2. https://colab.research.google.com/drive/1xDXYyQ_U__jMzjRYKG2L55D43hJ11u-l

[3] J22Melody. (n.d.). *SignWriting Translation* [Computer software]. GitHub. Retrieved [01-11-2024], from https://github.com/J22Melody/signwriting-translation

[4] Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2018). Neural Sign Language Translation. CVPR, 7784-7793.

[5] Yin, S., Xia, Z., Chen, X., Zhou, H., & He, S. (2020). Sign Language Translation with Transformer. MM '20, 1778–1786.

[6] Orbay, E., & Akarun, L. (2020). Neural Sign Language Translation by Learning Tokenization. arXiv:2004.03519.

[7] Camgoz, N. C., Hadfield, S., Koller, O., & Bowden, R. (2020). Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. CVPR, 10023-10033.

[8] Saunders, B., Camgoz, N. C., & Bowden, R. (2020). Progressive Transformers for End-to-End Sign Language Production. ECCV, 687–705.

[9] Pu, J., Zhou, P., Wang, F., & Xu, W. (2019). Boosting Continuous Sign Language Recognition via Cross Modality Augmentation. CVPR, 11509-11518.