Design of a Mobile App to translate speech to American and Peruvian Sign Language using an LSTM model

Joaquin Galván
Ciencias de la Computación
Universidad Peruna de Ciencias Aplicadas
Lima, Perú
u20181a010@upc.edu.pe

Joaquin Flores
Ciencias de la Computación
Universidad Peruna de Ciencias Aplicadas
Lima, Perú
u201810807@upc.edu.pe

Patricia Reyes
Ciencias de la Computación
Universidad Peruna de Ciencias Aplicadas
Lima, Perú
patricia.reyes@upc.pe

***Abstract*—Sign Language (SL) is a visually perceived natural language with just as many properties as the more common spoken language. It is characterized by the presence of various visual signs that convey meaning and emotions given the different facial expressions and hand gestures used. This paper aims to use Machine Learning technology in the realm of SL production by translating spoken words into gloss form, a way to write signs while maintaining their unique grammar structure. This would later be used to power a 3D animated character to produce the correct sign for each spoken or written sentence. Our objective is to continue existing efforts to ease communication between hearing-impaired and sighted individuals and to allow the social inclusion of hearing-impaired people in their daily lives. This paper covers the creation of a translation system that automatically converts English or Spanish audio and text into American Sign Language (ASL) and Peruvian Sign Language (LSP), respectively.**

***Index Terms*—Long Short-Term Memory, Natural Language Processing, Machine Learning, Sign Language Translation, American Sign Language, Peruvian Sign Language, Sign Language.**

I. Introduction

For a human being, communication is a means to express emotions and information, interact with other people, and help the community evolve. However, deaf and hard-of-hearing communities around the world encounter communication difficulties while engaging with the general population because there are few ways to compromise online material to suit their understanding. The majority of research concerns the groups' education, employment, and healthcare, which are all affected by this language barrier. Therefore, these communities are in need of inexpensive translation services to improve their lives [1].

According to the 2011 American Community Survey, about 3. 6% of people in the United States consider themselves deaf or have a serious degree of hearing loss. In 2011 that figure came to around 11 million people which, after 13 years, we can assume has only increased. Although the Americans with Disabilities Act (ADA) mandates that everyone must receive equal access in places open to the public, we are still far away from a time when nonhearing people can easily communicate with anyone in their day-to-day lives.

Sign language translation is a research field that has been investigated since the late 1980s [2]. However, most of that work deals with solutions for sign identification. Generating signs, the field of interest in our project, is a new branch of research that has some problems to solve, which means that the results are promising.

Various Machine Learning (ML) techniques have been used to generate metadata that are later used to create avatar animations of different sign languages. Current techniques with the most favorable and best results are Transformers and Recurrent Neuronal Networks (RNN), and we explore these options.

This paper is divided into the following sections. First, we review related work on sign language translation and various Machine Learning models in Section II. Then, we discuss relevant concepts and the data flow designed as our main contribution in Section III. Furthermore, we will explain the procedures performed and the experiments conducted in this early version presented in Section IV. In the end, we will show the main conclusions of the project and indicate some recommendations for future work in Section V.

II. Related Works

This section provides an overview of recent research in Sign Language studies, focusing on different languages worldwide and Machine Learning models. In the end, we will also cover our selection process for the model used.

On a national scale in Peru, research has been quite limited, where one of the most insightful works was from Bejarano et al. [3], developers of PeruSIL, a framework to build a continuous interpretation dataset of the Peruvian Sign Language (LSP for its acronym in Spanish). This novel project will be devoted to the actual scarcity of video-based datasets for sign languages, mainly due to the challenges found in recording and annotating native signers. The dataset prepared by this research team is the first multimodal LSP dataset of its kind, containing both audio and video data, including its transcripts and key-point-landmark annotations. This dataset was used for the training of a model for the recognition of sign languages, with a precision of up to 80.3%, making the first steps in improving accessibility and communication for the deaf community in Peru. The framework will be important for the forward-looking development of technologies that process sign languages and will also highlight the need for the participation of the deaf community in the dataset creation and refining process.

Meanwhile, on a worldwide scale, there have been a more substantial number of studies that have researched more languages and Machine Learning models. For more clarity, we have divided the international related works into its respective ML model, covering different languages and its respective SL.

A. *Transformers*

Saunders and company [2], propose a model based on progressive Transformers that allows translating text input into continuous 3D animations of the required poses. Part of their proposals includes methods for reducing the variation in the production of their models and a technique for predicting the size of an input based on the current progress of the translation. A notable conclusion is the better performance they achieved by translating directly from text to sign language compared to the translation that uses an intermediate glossary. They inferred that this indicates that it may not be necessary to limit the research to data sets that include a glossary.

Mosa et al. [4] developed a real-time Arabic avatar for the deaf-mute community by integrating a deep learning framework with an attention mechanism. Their method involved creating a custom dataset of 12,187 Arabic–Arabic sign gloss pairs and employing a transformer-based model with self-attention mechanisms to translate spoken Arabic into Arabic Sign Language (ArSL). This approach utilizes Bidirectional Encoder Representations from Transformers (BERT) to embed input text, ensuring that the neural network effectively captures essential context for accurate sign language translation. The translated text is then animated through a 2D avatar, visually representing ArSL to facilitate communication. The system was rigorously evaluated, achieving a training accuracy of 94.71% and a testing accuracy of 87.04%, demonstrating its efficacy in bridging the communication gap between the hearing and hearing-impaired communities. This innovative application not only enhances social inclusion but also provides a scalable solution to the limited resources in Arabic sign language interpretation.

B. *Recurrent Neuronal Network*

Stoll et al. [5] presented a model that used Neural Networks, computer graphics models, and the Sequence-to-Sequence method to translate text or audio into Sign Language using a video generated by the model itself. For the task of translating text into a sign, Recurrent Neural Network methods are used, featuring an attention module with the aim of storing information over time to achieve a more accurate translation.

Guo et al. [6] proposed a hierarchical-LSTM encoder-decoder model to avoid positioning translated words wrongly, something that usually happens when referring to a visual object in a sentence. They translate sign language into written text with the help of a 3D convolutional neural network that analyzes the input video and two more LSTM networks that work as semantic translators and facial data manipulators. Although this approach gives them great results, they still struggle to translate sentences with less common words.

Mittal et al. [7] proposed a framework for continued sign language recognition using a modified version of the LSTM architecture. They added a "Reset" gate inside the algorithm in order to segment the continuous stream of signs given, thus, having more accurate results by allowing the model to limit itself to short translations. This change allowed them to have an accuracy percentage of up to 89.5%.

C. *Generative Adversarial Networks*

Although Generative Adversarial Networks (GANs) are a recent contribution [8], there are already proposed solutions using this method for Sign Language generation.

Natarajan et al. [9] developed a model to generate photo-realistic videos from a pose expressed in a virtual skeleton. They use the generative and discriminative network present in GANs, in addition to a third network that serves to improve the images created by the generative network. Although the results do not represent a significant improvement over other video generation frameworks, they do provide a workflow that allows for the automation of sign representation for text-to-sign language translations.

Stoll et al. [5] introduce a novel method to generate sign language videos from text using a combination of Neural Machine Translation (NMT) and GANs. This approach uniquely integrates motion graphs with GANs, avoiding the use of traditional avatars, thereby achieving more realistic and fluid sign language animations. The methodology consists of translating textual input into a sequence of sign language poses using an NMT model, which are then fed into a motion graph to generate a corresponding pose sequence. These sequences serve as conditioning data for a GAN that synthesizes realistic sign language video. This dual-stage translation from text to glosses and then to video allows for the production of continuous sign language sequences without the need for heavily annotated datasets or complex avatar controls.

Now we will explain the details behind our selection of the Machine Learning model and describe the data sets used in a concise way.

Table I shows a comparison of the learning models previously seen in this section, with Criteria and its Impact; the Criteria are fields that we believe cover the most important aspects of the models, detailed as:

- Learning Resources: Dataset size and the time the model needs to complete its training phase, the lower the better.

TABLE I
COMPARISON OF LEARNING MODELS BY CRITERIA AND IMPACT

Criteria	Impact	Learning Model					
		Transformers		Long Short-Term Memory		GAN	
		Score	Average	Score	Average	Score	Average
Learning Resources	20%	3	0.6	4	0.8	4	0.8
Use in Similar Solutions	20%	2	0.4	5	1.0	2	0.4
Processing Speed	40%	4	1.6	3	1.2	3	1.2
Implementation Time	20%	3	0.6	5	1.0	2	0.4
Total	100%		3.2		4.0		2.8

- Use in Similar Solutions: Number of solutions, projects, and research in the field of sign language translation that use that model, the more the better.
- Processing Speed: The model's speed in processing data input and providing the result, lower times are better.
- Implementation Time: The time required for the model implementation process, the lower the better.

The Impact was calculated by comparing the Criteria with each other in terms of importance, consequently, Processing Speed has more value than the other Criteria for the selection of learning model. Next, we evaluated each learning model on all Criteria and assigned a Score from 1 to 5, then multiplied the score by the impact to get the average. Finally, we added all the models' averages to get the total for the model (higher is better). And, for our criteria and judgment during the evaluation, Long Short-Term Memory was the model with the highest average, thus, we decided to use a recurrent neural network approach with the use of LSTM, more specifically, in an encoder-decoder model.

Our main deciding factor is the type of dataset needed for this model and the scarce available datasets for LSP translations. This type of model requires a dataset of Spanish and English sentences paired with their respective gloss representation. For context, in our research, we found only one publicly available LSP dataset created by Berjarano et al. [10] which contains, although in very small quantities, video examples of LSP words, sentences, their Spanish text translations, and their gloss representation; which is quite robust in the signs and sentences covered. On the other hand, ASL has a wide variety of datasets, so it was easier to compare and choose the appropriate data for our project. We chose to use two publicly available and fairly recently updated data sets, the first created by Oscar Koller [11] who ”propose the first real-life large-scale sign language data set” and collects over 25,000 annotated videos. And the second dataset being from ASL Signbank [12] developed and maintained by researchers at the University of Connecticut and Gallaudet University.

III. MAIN CONTRIBUTION

In this section, we will explain the workflow of our proposed application, list the components that we will need to implement it, how data flows through the LSTM encoder-decoder model, and some important concepts needed to understand it.

A. Preliminary Concepts

Here, we briefly describe the most important concepts needed to understand our proposed model. We aim to translate speech to Sign Language, using Natural Language Processing via a Long Short-Term Memory model to learn and make the translation, being presented to the user in a 3D avatar created with Skeletal Animation.

1) Sign Language: Sign Language (SL) is a visually perceived natural language with just as many properties as the more common spoken language. SL is incredibly useful for people who have severe hearing impediments because the user only needs their hands and facial expressions to transmit information.

Like spoken languages, SL is not universal. According to The National Geographic Society (2024), there are around 300 different sign languages all over the world that are being used by more than 72 million people. Because of this, it is difficult for people who only know one SL to communicate with others once they are outside of their country.

2) Machine Learning: Machine Learning (ML) is a subfield of artificial intelligence that focuses on developing algorithms and statistical models that allow computer systems to learn from data and improve their performance on a specific task without being explicitly programmed. These algorithms build a model based on sample data, known as training data, to make predictions or decisions without human intervention [13].

3) Natural Language Processing: Natural language processing (NLP) consists of obtaining and manipulating information gathered from text or audio files. This is normally accomplished through a Machine Learning model that has been trained with extensive structured datasets filled with real-life examples.

4) Neural Networks: Neural networks are a type of Machine Learning model that uses the human brain as an example of its structure. Neural Networks consist of many layers of neurons that connect to each other. These neurons will pass information between themselves only if the ingested and manipulated data exceed a certain threshold. Once the data has gone through enough changes and layers, it comes to a final output layer of the network. We train this network to get an expected output depending on the model's input.

5) LSTM: Long Short-Term Memory (LSTM) is a variation of recurring neural networks. The main variation, as the name implies, is its ability to remember information that would otherwise be discarded. This extra information allows the model to learn dependencies between the given data. A feature like this is incredibly useful in spoken and text translations.

As mentioned previously, we opted to use an encoder-decoder approach to our LSTM model due to the need for varying outputs and their effectiveness with machine translation. This model works by having two LSTM networks work with each other. The first takes the input data from an input layer and processes it as they normally would, but instead of handing these data to the output layer, they pass them to the decoder module where it will again go through heavy processing, but now with a different set of weight and biases before being sent to the output layer.

6) Skeletal Animation: Skeletal animation refers to a technique that gives the user control over two important objects, skeleton and mesh, to control the look and movement of a 3D object. Skeletons are just sets of three-dimensional vertices that, depending on the object in mind, can be connected or even have a hierarchy between them. This hierarchy is used as a tool to make lower-level vertices react when a higher-level ”bone” moves. The mesh, on the other hand, is the external layer of the 3D object. To make the mesh move, animators link various parts of it to one or many bones so that the mesh will expand or compress depending on the specific move taken by the skeleton.

B. Method

Now, we will proceed to explain the main contributions related to the Machine Learning model and the mobile application proposed in this paper.

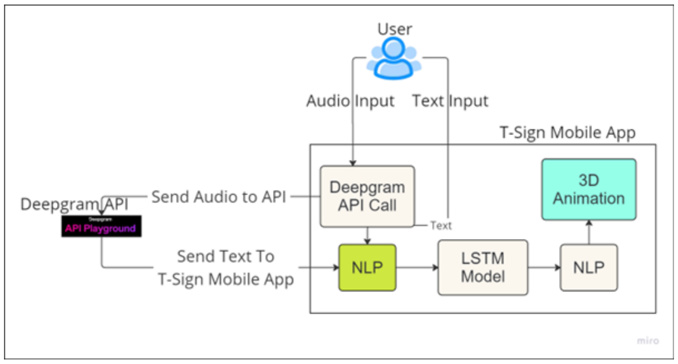


Fig. 1: A flowchart that shows the general aspects of our proposed program, own making.

First, the overall architecture of our application. As seen in Figure 1, the user will input text or audio into the app. If their input is audio, make a call to Deepgram's API in order to get the best transcription possible. After getting the text input from the API, we then use NLP methods to split the initial data into coherent 20-word sentences that we will then pass to our LSTM model. We then pass this sentence into our LSTM model to predict its gloss output. With the gloss text sentence in memory, we then match the words to their corresponding sign with the help of an NLP method that gives us the existing synonyms of any given word, more on this later. Finally, we show our user the translated sentences by making our 3D avatar sign every word needed. Given the simplicity of our app, we do not require additional databases or external API connections other than our connection to Deepgram.

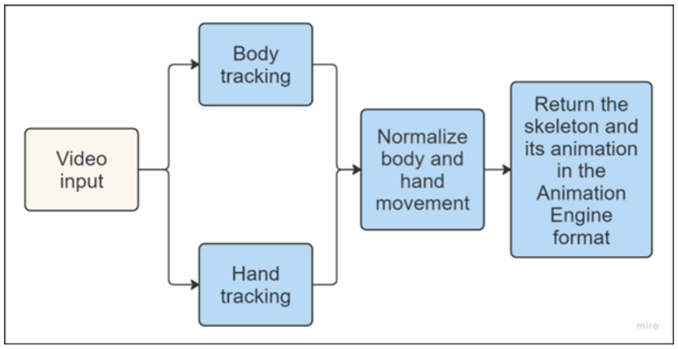
Now we will give a more in-depth, step-by-step explanation of our application's workflow.

After getting the user input and the response from Deepgram's API, we use the Natural Language Toolkit (NLTK) library in Python to tokenize and divide our input sentences. Additionally, we try to identify the names of people and companies in the sentence, as most of them will require finger-spelling for their proper translation.

For the Machine Learning model, we decided to use a recurrent neural network, specifically LSTM, due to its capability of holding important information between many iterations and its dataset requirements. As mentioned previously, we will need a collection of English/Spanish sentences with their gloss counterpart; this can also be called corpus. The memory advantage over common recurrent networks allows the model to keep order in the data once it has been transformed, which is incredibly useful to us as in order to maintain the meaning of a sentence, we need to be aware of the dependencies that exist between words. Although regular recurrent neural networks do have the capability of maintaining some information, LSTMs were designed specifically to maintain the relevant data for longer periods of time to avoid dependency issues.

Gloss allows us to give a specific sign a word (or words) to identify it. Our LSTM model would then transform the natural text given to it into a gloss equivalent. This new gloss ”sentence” should have the proper grammatical order used for sign language and should only keep the relevant words. To train our model, we will use gloss sentences and their English/Spanish language counterparts taken from actual real-life examples and examples available online. The data sets used for gloss sentences are LSP-PUCP 305 [10] for LSP and ASLG-PC12 [14] for ASL.

In order to get a 3D avatar to sign our translations, we first have to extract the motion and hand poses out of our 2 SL datasets and then apply that movement to our avatar when needed. Figure 2 shows the steps designed to create the avatar based on the videos from the data sets. Our video data comes from the MS-ASL American Sign Language Dataset from Microsoft [11], the ASL Signbank [12], and the previously mentioned LSP-PUCP 305 [10]. First, we used the existing libraries to obtain the skeleton and movement of each video. However, many whole-body skeleton detection algorithms do not give us detailed hand movements, so we are obligated to use a separate library to track the hand movements



Video input
Body tracking
Hand tracking
Normalize body and hand movement
Return the skeleton and its animation in the Animation Engine format
miro

Fig. 2: A flowchart that shows the steps to create a 3d avatar from videos, own making.

specifically and then match both skeletons together. For hand tracking specifically, we use Google's Mediapipe and OpenCV libraries in Python. Once we have the two skeletons and their movements, we can now apply them to a 3D avatar of our choice. Each sign animation contains a label that identifies it. Once we have a translated gloss sentence, we match the words in the sentence with the labels for each movement and stack them in a queue so they can be shown to the user in order. It is very possible that, with the limited-size dataset that we have for LSP, we may encounter some words that do not have an exact video to match them. To avoid problems when matching gloss words to their video counterparts, we use the NLKT library to find the closest word possible to the given gloss word. This means that sometimes the model will show a synonym rather than the exact word.

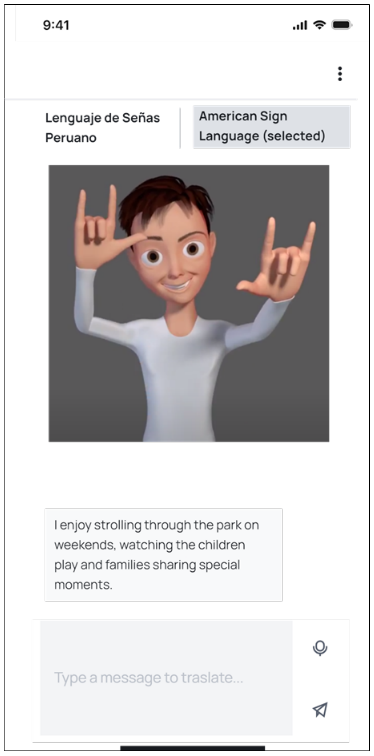
To conclude with this section, we want to show a preview of the proposed end-user product. Figure 3 shows the main interface of interaction in the application, where it lets the user select the sign language desired; this step is important because the model expects the input to be in the correspondent language, Spanish for LSP and English for ASL. Then in the middle of the screen is the 3D avatar created that displays the translated translation returned, being the main focus of the mobile application it uses the most space in the screen. Lastly, at the bottom, is the text sent to translate and the section to input the data, text or audio.

IV. Experiments

In this section, we will discuss the platform, databases, and inputs used in our experiments, as well as some initial results given by the first version of our LSTM model.

However, before continuing, it is important to know that to train our model we need our input data as well as an expected outcome. For our work, this means we will feed our model English/Spanish text, and it will learn to transform it into a gloss sentence.

Regarding where we trained and executed our mode, our experiments were run locally on a platform with an AMD Ryzen 7 5800X3D @ 4.5GHz, 32GB of RAM, and a Radeon 5700. The code was written and executed in Python 3.12.3



9:41
Lenguaje de Señas Peruano
American Sign Language (selected)
Type a message to traslate...

I enjoy strolling through the park on weekends, watching the children play and families sharing special moments.

Fig. 3: Wireframe of the proposed mobile app, own making.

where the most important libraries were Keras, Tensorflow, Numpy, and Pandas.

Continuing with our databases, while both gave us paired sentences, they are shaped very differently. The ASLG-PC12 dataset consists of two CSV files, one for the ASL sentences and another for the English pairs. These files are easy to extract as both of them consist of only 80 thousand sentences divided into a line each. On the other hand, the LSP-PUCP 305 is comprised of 130-ish folders that represent every sign in the database. Each of these folders has one or more videos of the corresponding sign, a video of a sentence example, and a .eaf file for every video. These .eaf files come from the ELAN software [15] and are based on the XML file structure. Every .eaf file contains either the gloss to represent the corresponding sign or both the Spanish sentence and gloss translation of the example sentence.

Now we will explain how we prepare the input of our model as well as its structure. After extracting the pair sentences of each dataset, we noticed that the annotations used to represent specific motions in gloss sentences can be a distraction for the model. Let us take for example the sentence "I am Jules". Following the gloss representation format used in ASLG-PC12, the new sentence would be "X-I Jules". This added "X" represents a finger pointing somewhere, in this scenario, it is towards Jules. Given its very contextual use, we decided to remove this kind of annotation from the input for the model. We do this before removing all punctuation from the sentences and after changing every character to lowercase.

Once we have the cleaned sentence, we proceed to tokenize our datasets. Tokenization refers to the separation of each sentence into unique words and then the assignment of an index value to each word. This allows us to present our model with numerical values with which it can work. In addition, we pad every sentence. This means that we add zeros at the end of our now tokenized array. We do this to match the length of the longest sentence in our dataset, as our model expects the same input lengths.

Now, our padded and tokenized sentences are ready to be used as input for our model.

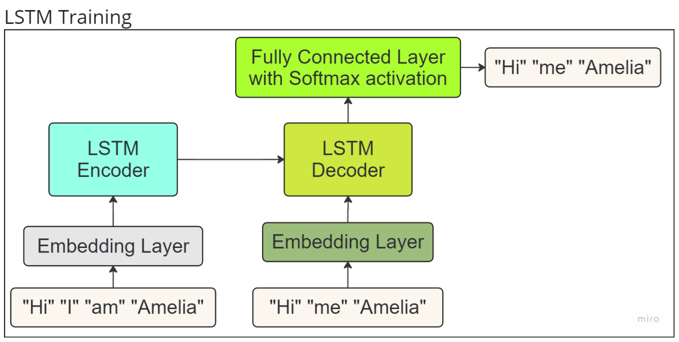


Fig. 4: A graphical explanations of the the LSTM encoder-decoder model

Figure 4 illustrates the LSTM training. It begins with an input sequence (”Hi”, ”I”, ”am”, ”Amelia”) that is converted into a dense vector representation via an embedding layer. These vectors are processed by an LSTM encoder that unrolls the information into a context vector. This context vector is then fed into an LSTM decoder, which, with the help of its own embedding layer, to correct the weight and biases of the decoder if the output is wrong, generates the output sequence (”I”, ”Amelia”) one word at a time. Each decoder output passes through a fully connected layer with softmax activation to produce a probability distribution over the tokenized vocabulary, ensuring the most probable next word is selected, resulting in the final translated or predicted sequence.

The most important parameters in our training model are the following: Batch size: 512, epochs: 20, the ”Adam” optimizer, and a learning rate of 0.001 The shown examples are from our work with the ASL dataset.

In Figure 5 we can see that the training in our model reports a loss value of 0.902 This is most likely due to the dataset being heavily annotated as well as it not being a direct

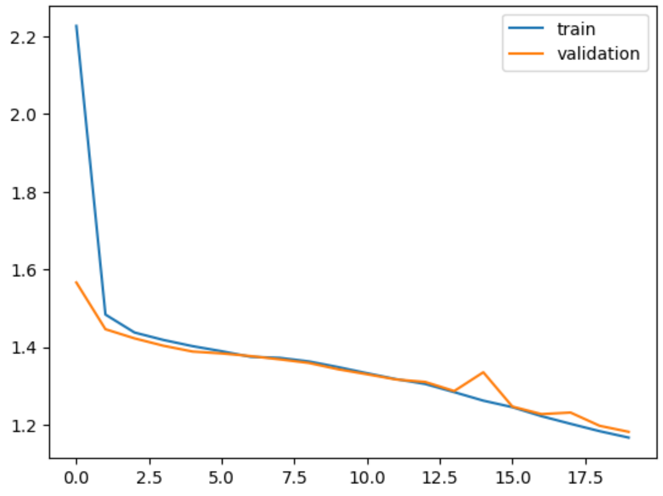
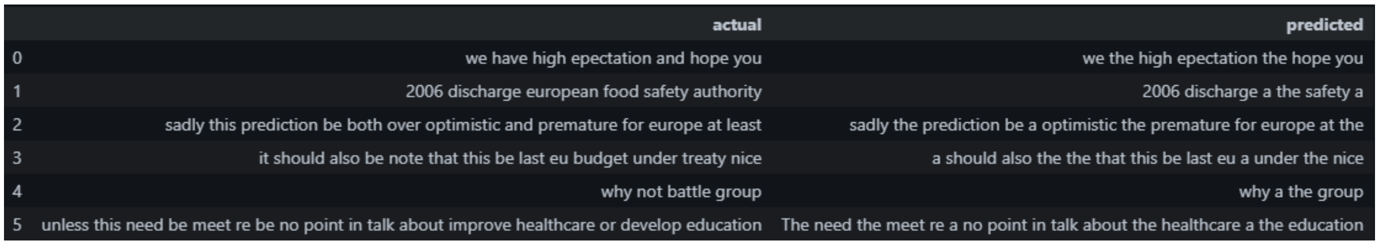


Fig. 5: A graph of our loss value according to the Epoch.

translation of every day use sentences. The available dataset contains sentences that often lack context or proper grammar structure. Because of this, we expect very limited success in achieving our translation with this dataset.

As we can see in Figure 6, there are many errors in the predicted out of our model, but it still manages to correctly translate some of the words. Although changes were made to the model parameters, these results lead us to believe that there are better settings for our specific model.

While we are grateful for the existing data sets, especially when there is so little of them, we do observe two major problems. In our ASL dataset, we can see that it was taken from a transcript of a government entity. Because of this, most of the samples are sentences that are either repetitive or do not represent a real conversation someone could have. The LSP dataset, on the other hand, does not suffer from this issue, as they made a special effort to not only use real-life examples, but also employ experts in LSP translations to get accurate translations. However, this extra effort could help explain the problem with this dataset. There are less than 300 examples for our model to use, which means that training is incredibly limited.



	actual	predicted
0	we have high epectation and hope you	we the high epectation the hope you
1	2006 discharge european food safety authority	2006 discharge a the safety a
2	sadly this prediction be both over optimistic and premature for europe at least	sadly the prediction be a optimistic the premature for europe at the
3	it should also be note that this be last eu budget under treaty nice	a should also the the that this be last eu a under the nice
4	why not battle group	why a the group
5	unless this need be meet re be no point in talk about improve healthcare or develop education	The need the meet re a no point in talk about the healthcare a the education

Fig. 6: The first six predicted results from our model on the right side, and the expected output on the left side.

V. Future works

As we have seen in our experiments, we suffer greatly from a lack of properly translated gloss sentences in our ASL datasets and a small number of samples in the few existing datasets for LSP. We believe that with more robust datasets made by experts in Sign language translation, models such as ours will have much better results, and with the new data available more work can be done in the area, specifically in Peru, where research regarding LS production and recognition is still in it's beginnings.

Also, as seen in this paper, we have only treated the design in terms of the learning model and the mobile app, so it remains the work, especially for us, to apply this proposal and a greater variety of results. But we encourage more people to research and develop the field of Sign Language Translation more, because we firmly believe that bringing these kinds of tool will close the gap between deaf and non-deaf people.

References

[1] A. Núñez-Marcos, O. Perez-de-Viñaspre, and G. Labaka, "A survey on sign language machine translation," *Expert Syst. Appl.*, vol. 213, no. Part, p. 118993, 2023.

[2] B. Saunders, N. C. Camgöz, and R. Bowden, "Progressive transformers for end-to-end sign language production," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI* (A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, eds.), vol. 12356 of *Lecture Notes in Computer Science*, pp. 687–705, Springer, 2020.

[3] G. Bejarano, J. Huamani-Malca, F. Cerna-Herrera, F. Alva-Manchego, and P. Rivas, *PeruSIL: A Framework to Build a Continuous Peruvian Sign Language Interpretation Dataset*. European Language Resources Association (ELRA).

[4] D. T. Mosa, N. A. Nasef, M. A. Lotfy, A. A. Abohany, R. M. Essa, and A. Salem, "A real-time arabic avatar for deaf–mute community using attention mechanism," vol. 35, no. 29, pp. 21709–21723.

[5] S. Stoll, N. C. Camgöz, S. Hadfield, and R. Bowden, "Text2sign: Towards sign language production using neural machine translation and generative adversarial networks," *Int. J. Comput. Vis.*, vol. 128, no. 4, pp. 891–908, 2020.

[6] D. Guo, W. Zhou, H. Li, and M. Wang, "Hierarchical LSTM for sign language translation," vol. 32, no. 1.

[7] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A modified LSTM model for continuous sign language recognition using leap motion," vol. 19, no. 16, pp. 7056–7063.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," *CoRR*, vol. abs/1406.2661, 2014.

[9] B. Natarajan and R. Elakkiya, "Dynamic GAN for high-quality sign language video generation from skeletal poses using generative adversarial networks," *Soft Comput.*, vol. 26, no. 23, pp. 13153–13175, 2022.

[10] M. Rodríguez-Mondoñedo, J. Pérez-Silva, H. Velasquez, S. Oporto, F. Cerna, C. Ramos, C. Vasquez, and G. Bejarano, "Lengua de Señas Peruana (LSP) - PUCP 305 (glosas)," 2024.

[11] O. Koller, "Ms-asl," Aug 2019.

[12] J. A. Hochgesang, O. Crasborn, and D. Lillo-Martin, "ASL Signbank." https://aslsignbank.haskins.yale.edu/, 2017–2024. New Haven, CT: Haskins Lab, Yale University.

[13] R. Kenge, "Machine learning, its limitations, and solutions over it," *International Journal of Information Technology Modeling and Computing*, vol. 11, pp. 73–83, 10 2020.

[14] A. Othman and M. Jemni, "Designing high accuracy statistical machine translation for sign language using parallel corpus: Case study english and american sign language," *J. Inf. Technol. Res.*, vol. 12, no. 2, pp. 159–174, 2019.

[15] X. Zhang, H. Zeng, S. Guo, and L. Zhang, "Efficient long-range attention network for image super-resolution," 2022.