ASL2TEXT

Ferran Vera Filella

AI

Introduction

The intersection between artificial intelligence and language translation has seen significant advances in recent years, revolutionizing the way we communicate between diverse communities. In this project, I present a comprehensive exploration and implementation of an innovative system aimed at transforming American Sign Language (ASL) into text. This innovative effort is an example of the power of AI to improve accessibility and inclusion for the hearing impaired. Throughout this project, I delve into the ins and outs of our ASL-to-text translation system, describing its development process, the methodologies employed, and the challenges encountered.

Literature Review

American Sign Language (ASL) is a complex and expressive visual language used predominantly by the deaf and hard of hearing communities in the United States and parts of Canada, possesses its own grammar, syntax, and semantics. Its evolution, linguistic structure, and regional variations have been extensively studied, highlighting the need for nuanced translation systems that capture the subtleties of this visual language.

Advances in computer vision involve sophisticated algorithms capable of interpreting visual data, crucial for understanding sign language gestures. These systems use machine learning models to learn and identify patterns from visual data, making it possible to interpret sign language movements.

These sign language recognition systems are designed to use cameras as input devices, capturing the movements and gestures inherent in sign language communication. Using machine learning algorithms, these movements are interpreted and translated into written or spoken language.

Data processing

For this project I have used a dataset provided by Kaggle[1] “Google - American Sign Language Fingerspelling Recognition”. The ASL Fingerspelling Recognition Corpus is a collection of hand and facial landmarks generated by Mediapipe on videos of phrases, addresses, phone numbers, and urls fingerspelling by over 100 Deaf signers. This dataset includes fingerspelling of letters, numbers, and symbols at real world speeds. This may help move sign language recognition forward, making AI more accessible for the Deaf and Hard of Hearing community. The data was stored in parquet files, so kaggle provided us[2] a file to transform files from parquet to TFrecords. The transition to TFRecords was driven by two primary advantages. Firstly, TFRecords exhibited a marked improvement in data processing speed, enhancing the overall efficiency of our pipeline. Secondly, the TFRecord format facilitated a more seamless and efficient handling of both landmark and phrase data within our workflow. This strategic shift from Parquet files to TFRecord format stands as a critical methodological choice, optimizing data organization and processing, thus laying a robust foundation for the subsequent phases of our project.

Feature Selection and Processing

Within the dataset I identified specific columns corresponding to X, Y, Z coordinates, and relevant labels associated with pose and hand gestures. Through careful selection, columns pertinent to hand and pose-related indices were chosen, enabling the extraction and processing of essential features required for subsequent modeling.

Model architecture

The most important part of our ASL-to-text translation system lies in the adoption of the **Transformer** architecture, a robust model known for its high success rate in processing sequential data. At its core, this architecture employs an **encoder-decoder** framework, a fundamental structure that enables the understanding and translation of ASL gestures into coherent textual representations.

Within this architecture, the **encoder** serves as the initial processing unit for the input ASL landmarks. It operates through a series of layers, each comprising distinctive sub-layers. These sub-layers encompass a **multi-head self-attention mechanism** and subsequent **feed-forward** neural networks. The **multi-head self-attention** fosters comprehensive interaction among diverse segments of the input sequence, allowing the model to discern intricate relationships within the ASL landmarks. This mechanism is complemented by **feed-forward** neural networks, adding depth and non-linearity to the encoded representations, further enriching the model's understanding of the input.

Conversely, the **decoder** component of the architecture is tasked with generating textual representations based on the encoded ASL landmarks. Similar to the encoder, the decoder incorporates multiple layers housing essential sub-layers, including **self-attention** and **encoder-decoder attention** mechanisms. The **self-attention** capability enables the decoder to focus on different facets of the output sequence during the translation process, ensuring generation of textual tokens. Additionally, the encoder-decoder attention facilitates the decoder's concentration on pertinent aspects of the encoded input (ASL landmarks), facilitating contextually accurate translations.

The Transformer architecture addresses the inherent lack of sequential information retention by incorporating positional encodings to impart crucial sequence order information. **Token embeddings** represent textual tokens as numerical vectors, while **landmark embeddings** process raw ASL landmarks into suitable formats for the model's comprehension, ensuring the model's ability to comprehend and translate complex ASL gestures effectively.

During the training phase, our system minimizes a defined loss function, categorical cross-entropy with label smoothing, utilizing the Adam optimizer to optimize the model's parameters. Paired ASL landmark sequences and their corresponding textual representations comprise the training data, enabling the model to learn accurate translation patterns from gestures to text.

The incorporation of attention mechanisms within both the encoder and decoder components is pivotal to our system's success. These mechanisms allow for comprehensive dependencies within the input sequence for the encoder and generation of output sequences in the decoder, considering both previously generated tokens and input landmarks.

The Transformer architecture, with its attention-based design and parallelizability, stands as the backbone of our ASL-to-text translation system. Its efficiency and efficacy empower the precise and efficient translation of ASL gestures into their corresponding textual representations, marking a significant stride in bridging communication barriers for the hearing-impaired community.

Experiments

Training and Optimization

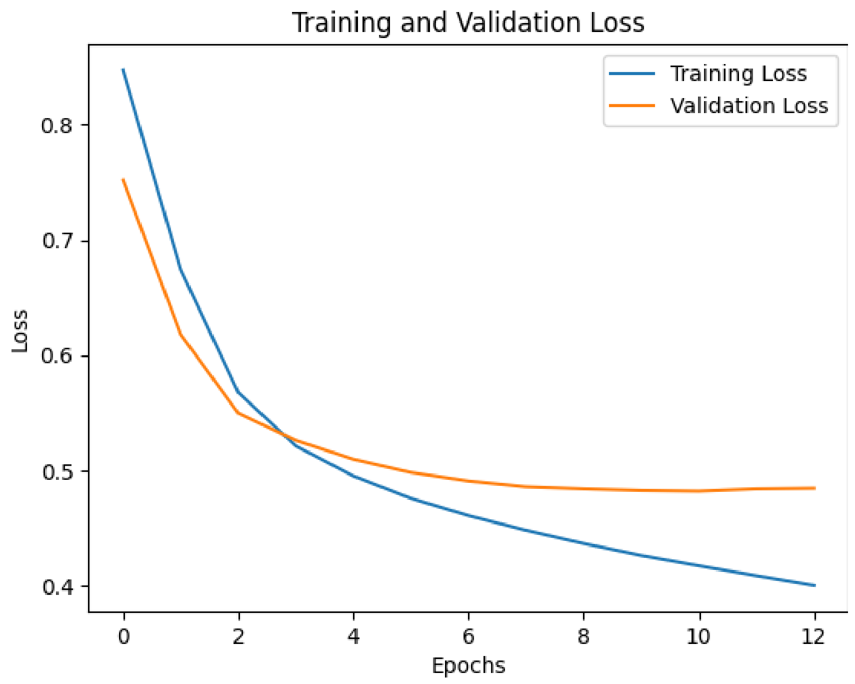
Before initiating the training, the dataset was meticulously split into distinct subsets: a training set, a validation set, and a test set, with 60%, 20% and 20% of the total dataset respectively. The training dataset consists of paired ASL landmark sequences and their corresponding textual representations. This dataset serves as the foundation for the model's learning process, allowing it to grasp and internalize the intricate correlations between ASL gestures and their textual equivalents. Through this process, the model gradually hones its translation patterns, learning to accurately convert gestures into meaningful text. Simultaneously, the validation set aids in monitoring the model's learning progress, ensuring it doesn't overfit to the training data by evaluating its performance on unseen samples. The test set, unseen during training, serves as the ultimate evaluation benchmark, assessing the system's real-world applicability and generalizability.

The optimization mechanism is fundamental to this learning process. Leveraging the Adam optimizer, our system meticulously adjusts the model's parameters during training to minimize the defined loss function. This optimizer, known for its efficiency and adaptability, enables the model to navigate through the complex landscape of ASL landmarks and textual representations, gradually improving its translation accuracy.

Label smoothing, a regularization technique, plays a pivotal role during training. By introducing slight uncertainty into the true labels of the training data, label smoothing prevents the model from becoming overconfident and overly reliant on specific patterns. This strategy encourages the model to explore a broader spectrum of translation possibilities, fostering a more robust and adaptable translation capability.

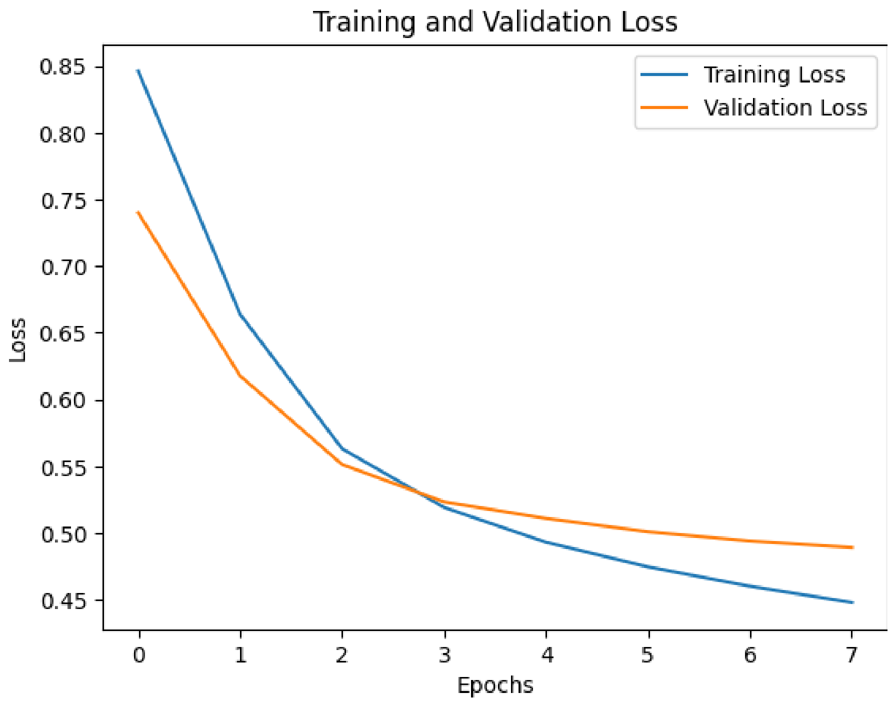
Results

The Model Architecture section provided insights into our application of a Transformer-based framework characterized by 2 encoder layers and 1 decoder layer. Each encoder layer incorporated a multi-head self-attention mechanism and a feed-forward neural network, while the decoder layer featured self-attention and encoder-decoder attention mechanisms. This configuration enabled our model to effectively comprehend the nuances of ASL gestures and generate accurate textual representations. Key Transformer model hyperparameters included a hidden dimension size of 200, employing 4 attention heads, a feed-forward network dimension of 400, with 2 encoder layers and 1 decoder layer. I utilized categorical cross-entropy with label smoothing (0.1) as the loss function and the Adam optimizer with a learning rate set at 0.0001. During training, meticulous tuning of these parameters and strategies allowed to mitigate overconfidence in the model. Employing a batch size of 32 facilitated efficient processing, and our training regime spanned 13 epochs, empowering the model to progressively refine its translation capabilities.



As you can see in the previous image, the model learns, but from epoch 8 onwards you can see how the training loss keeps going down but the validation remains constant, this means that the model is overfitting, so the next experiment would be the same but training the model for 8 epochs.

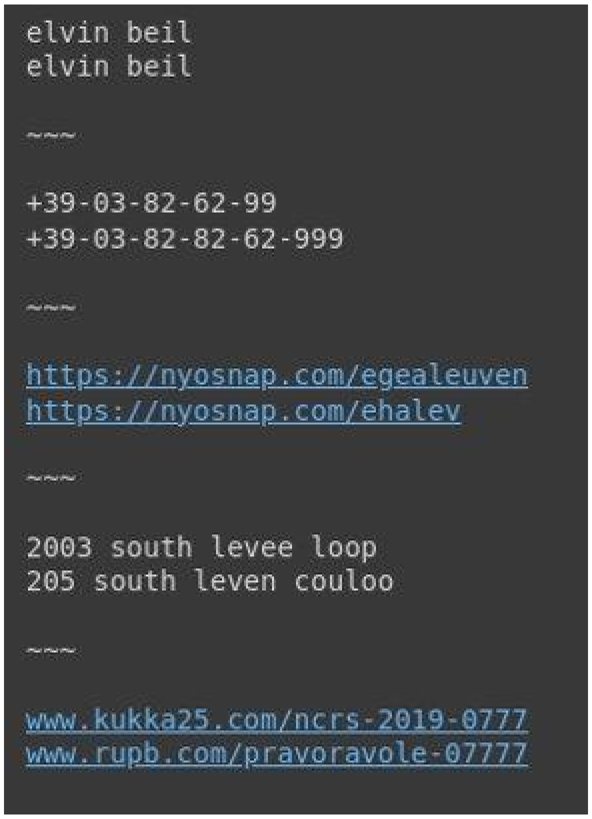
The image below shows the model's training process using the same hyperparameters but changing the number of epochs, now instead of training the model for 13 epochs we do it for 8 to try to prevent the overfitting. As you can see there is not as much separation between training loss and validation loss as with 13 epochs.



The evaluation metric utilized for this project is the normalized total **Levenshtein**[3] distance. To elaborate, let's denote the total number of characters present across all labels as *N* and the total Levenshtein distance calculated as *D*. The metric is formulated as follows: (N - D) / N.

This metric essentially quantifies the dissimilarity between predicted and reference texts. *N* represents the total number of characters in all ground truth labels, while *D* signifies the total Levenshtein distance, measuring the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform predicted text into the reference text.

The accuracy obtained when the model was trained with 13 epochs was 66.88%, which is quite good, in the following image we can see an example of prediction, the first row is the true phrase and the second row is the predicted. When the model was trained with 8 epochs, to prevent what I thought it was overfitting, the accuracy obtained was
61.50%.



```
elvin beil
elvin beil

~~~

+39-03-82-62-99
+39-03-82-82-62-999

~~~

https://nyosnap.com/egealeuven
https://nyosnap.com/ehalev

~~~

2003 south levee loop
205 south leven couloo

~~~

www.kukka25.com/ncrs-2019-0777
www.rupb.com/pravoravole-07777
```

Prediction Example

Challenges and Future Work

In developing this sign language interpreting system, I encountered significant challenges. One of the main obstacles I faced was the complexity of finding a quality dataset that would allow me to develop the project.

In addition, the duration required to train the model proved to be another major challenge. Despite taking advantage of powerful GPU resources, using google colab, and using a sizable data set, training our model for 13 epochs still required approximately 1 hour.

Looking to the future, this project opens the door to several avenues of work and improvement that could revolutionize sign language interpreting technology. One of them is the development of a real-time translation system. The creation of a platform capable of interpreting sign language gestures instantaneously in real-life situations could be a game changer and greatly facilitate communication and accessibility for the deaf and hard-of-hearing communities in real-life environments.

In addition, it is interesting to explore the possibilities of bidirectional translation. Extending the functionality of the system to translate from text and video formats into American Sign Language (ASL) would greatly enhance inclusion. This advancement would not only enable seamless communication from ASL to other linguistic forms, but would also facilitate translation in the opposite direction, thus bridging communication gaps for people using different modes of communication.

Conclusion

In conclusion, developing this project has been a very important experience because I have been able to learn more about the deaf and hard-of-hearing community. By overcoming challenges such as the complexity of the hand gestures, the complexity of the data, and the length of the model training, I have gained valuable insights into the multifaceted nature of sign language interpreting.

Looking ahead, the road ahead is lit by promising opportunities for innovation. The prospects of real-time translation and two-way communication represent horizons yet to be explored. The vision of a system capable of interpreting sign language gestures in real-life situations and facilitating translation between different modes of communication opens up a range of possibilities for social impact and inclusion.

References

[1] https://www.kaggle.com/competitions/asl-fingerspelling

[2] https://www.kaggle.com/code/shlomoron/aslfr-parquets-to-tfrecords-cleaned

[3]https://www.cuelogic.com/blog/the-levenshtein-algorithm#:~:text=The%20Levenshtein%20distance%20is%20a,one%20word%20into%20the%20other.

[4]https://imatge.upc.edu/web/sites/default/files/pub/xCabot22.pdf

[5] https://aclanthology.org/2020.coling-main.525.pdf

[6]https://www.researchgate.net/publication/370215194_Artificial_Intelligence_for_Sign_Language_Translation_-A_Design_Science_Research_Study