Final Project Report : Aligning Text to Sign Language Video

Elysheva Dray
Université Paris Dauphine - PSL
MASH
elycheva.dray@dauphine.eu

Samuel Marciano
ENS Paris Saclay
MVA
samuel.marciano@ens-paris-saclay.fr

Abstract

In this final project work, we first reproduce the results of a previous work [3], in order to align a text from a sign language video. Our main model was a transformer and once we handle enough this work, we attempted to improve it and have better result than the reference given.

1. Introduction

Sign languages are visual languages that have evolved in deaf communities. Currently, translation systems for sign languages are not really efficient. For instance, although sign language interpreted TV broadcasts are readily available, the subtitles are usually aligned with the audio rather than the sign language. (see. Figure 1)
In order to develop models capable of improving the translation and transcription of signs in a sign language video, BBC provides the BOBSL dataset. This dataset is used throughout this project aiming to work on **aligning** text sentences to the correct location in sign language video. For this, we will try step by step to reproduce baseline results of the paper, and to improve the model using the architecture base of the SAT model.

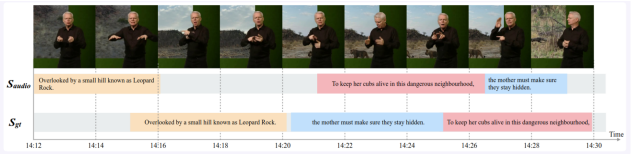

Figure 1. Idea of the project purpose, from [5]

2. Dataset description

2.1. Global description

For this part, we introduce BOBSL dataset and we briefly explain how it has been built so we can use it in the best possible way for sign language translation and transcription tasks.
The access to the BOBSL dataset has been given to us and we observe various files that compose it: namely the videos, and the subtitles.
All these data help us to perform the task in question. But to get this data, we had to pre-process them. Here are some examples of actions done on this dataset before we access it.

First of all, it has been processed in such a way that we can concentrate on interpreters, meaning we have cropped the images to focus on the signers.
Then, the faces that could appear in the background images have been blurred and information about signers' faces have been collected in order to detect them easily.
Finally, the dataset has been split in different sets: train, validation and test according to the signers: 28 signers from 1658 videos for the train set, 7 signers from 32 videos for validation set and 4 signers from 250 videos for the test set. For more details, see [1].

Moreover, this dataset contains not only images/videos but also subtitles. It has therefore been partially annotated. We note 13 hours of strong annotations and nearly 1000 hours of weak annotations, both on the train set.

2.2. Annotations

Now, we will try to deal with a main problem in this paper, on how we annotate the BOBSL dataset in order to have a proper dataset ready to train. We can imagine that annotate manually the data can be very costly on time, money and knowledge. That's why, in order to avoid this problem, we have 2 different types of annotations : one manual and one automatic. (see section 3 of [1])

2.2.1 Manual annotations

The 13 hours of strong annotations refer to the manual ones. We count 16 manually aligned videos in the train set, 4 manually aligned validation videos and 35 ones from the test set. We use two manual annotation types :

- Sign verification by informing the veracity of a sign in British Sign Language.
- Sentence Alignment we adjust the time to the list of sentences

2.2.2 Automatic annotations

We have 1000 hours of weak annotations that refer to automatic annotations. For this kind on subtitles, we have 2 main different ways to process.

- The mouthing keyword spotting approach : we consider also the movements of the head and facial expression. The main role of the mouthing is to exclude case of homonyms, and put a marker for sign spottings. With a word in input, we search the mouthing pose corresponding in a window of 10 seconds before and after the listening audio of the current word.
- Dictionaries: we want to measure similarity between isolated dictionary videos and continuous signing. By averaging features, we obtain a single embedding for the dictionary sample. We aim to record the location where the similarity is maximized. So we choose the best match as the one with highest similarity score.

3. Method

Now we will present the methods we will use and which model. We have two different models, one for the recognition of the signs pose and one to align the sentences. (see section 4. of [1])

3.1. Sign recognition model

First for sign recognition, we use a 3D conventional model called i3d model from Google [4]. Poses are extracted using OpenPose, hence we can have the sign according to the pose. We train it on all the BOBSL dataset, with 5 epochs, stochastic gradient descent with momentum = 0.9, batches of 4 videos, In order to have some data augmentation, we add some color augmentation, scale and horizontal flip.

3.2. Sign language sentence alignment model

We now present our main model, which predicts what we are looking for. First, we use a Transformer model : we locate a text query corresponding to a sentence in a certain window. In the input of the encoder, we have BERT token embeddings of the text query we wish to align. In the decoder's input, we have the video features extracted from a i3d model, presented in subsection 3.1.

After pre-training the SAT model, we finetune using train set and spottings filtered, and we finetune also on human aligned train set (manually aligned). Finally we train with all our set. For data augmentation purpose, we shuffle 50 % of the sentences and drop 15%.

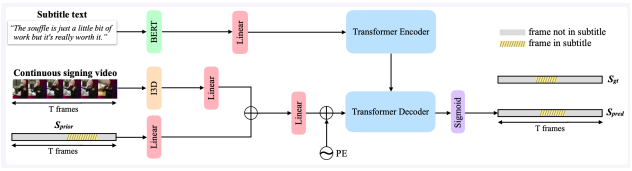

Figure 2. SAT model overview, [2]

4. Our work

4.1. First steps

Before executing any part of the provided code, we read the procedures and try to retrieve the data and download the features we are going to use first. So the very first step of our project was to request access to the BOBSL dataset from BBC. Once this access was granted, we tried to work locally, but the too large amount of data and the need to use a GPU led us to turn to a virtual machine. So we decided to use a (GCP) Google Cloud Platform virtual machine.
After many tests, we managed to find the parameters of the virtual machine that we needed, namely: an important RAM (104GB), a large number of GB to store the data (400 GB) and a GPU to optimize the execution time and training of the model (Nvidia T4 or Nvidia V100). After the creation of our VM, we imported our data: the video features, the subtitles, the annotations and the spottings we needed. Finally, the environment was ready to be executed, but we still had to be careful about what we were executing and how we were using our data to train the model so that the VM would not trip again and again.

4.2. Reproducing baselines results

To train the model, we can't use the videos it-selves because it is too loud to process with them. Thus, we work only with their I3D features extracted. For this, we unzip only the files corresponding to the manually-aligned annotated videos features to work with. To get some basic results and visualize what we are doing here, we decided to only run the *finetune.sh* file on the manually-aligned subset of BOBSL. We used the checkpoints of the reference model [3]. We modified some parameters, the batch size from 64 to 32 and the workers from 32 to 16. And then, using the model we gave us, we test it on the manually aligned test videos which are the our ground truth. We finally obtain our first results which coincide with the reference ones.

4.3. Train SAT on dataset's subset

Now that we had something working, we decided to extend to the weakly aligned audio subtitles, used as a prior

location for localising the signing-aligned subtitles and we run the pre-training steps. As the amount of data is important, we chose to work with a bit less that 10% of each train, validation and test sets:

- 100 train videos randomly picked from the 1658 in the train set
- 5 validation videos randomly picked from the 32 in the validation set
- 15 test videos randomly picked from the 250 in the test set

After unzipping the files corresponding to the randomly picked videos, we follow the next procedure (described in [3]):

1. word_pretrain : *$ bash word_pretrain.sh*, to pretrain the model for the spotting detection. The model learns to localise sign spottings from mouthing and dictionary annotations.
2. train : *$ bash train.sh*, We switch to real subtitles, by considering the ground truth signing-aligned subtitles (heuristic corrected audio-aligned subtitles that have been shifted by +2.7s) and we train our model on it using the 100 videos we chose right before.
3. finetune : *$ bash finetune.sh*, to train only on the manually-aligned subset of BOBSL.
4. test : *Bash test.sh*, to evaluate our models and produces Video Text Tracks files with the alignment of our text depending on the time.

Each of those steps have different parameters to discuss before running the procedures. We choose for each one of the train to reduce the batch-size at 32 and the $N_{workers}$ at 16. We also reduced the amount of shuffle words to 45% and increase the dropping of words in a sentence to 20%. Also, we trained on 100 epochs for the *word_pretrain*, 90 epochs for the *train* and 100 epochs for the *finetune*. We kept the learning rate at 1.10^{-6} for the *finetune*, 5.10^{-6} for the *train* and 1.10^{-5} for the *word_pretrain*.

5. Our results

5.1. Quantitative results

Finally, after more than 48h hours of training in Nividia (T4 or V100 according to the availability), we obtain those results Table 1.

We computed over 2642663 frames, 20338 sentences. We have to notice that, we only train on less than 10% of the dataset whereas the reference of the SAT model is trained on more than 1600 features video. We can say that those results are satisfying regarding the amount of data we have.

Method	Frame-acc	F1@.10	F1@.25	F1@.50
SAT [Ref]	70.37	73.33	66.32	53.18
Our test	68.96	71.83	64.32	50.47

Table 1. Results

5.2. Qualitative results

We try to illustrate our results on alignment text to sign language from video. We pick two videos, where we selected one amount of time and we compare upon S_{audio}, which is the real time audio we listen from the speaker on TV, the S_{model}, that we predicted to align with our model and S_{GT}, the ground truth, which is the real alignment done by human (manually aligned).

Figure 3

6. Improve the model

To improve the model, we wanted to focus on the transformer architecture and modify it a bit by adding some information about the automatic sign spottings in the transformer. To do so, we would like to input them in the same way the "prior" is inputted to the decoder as in Figure 2. Transformer-based model aims, here, to locate text subtitles in a window of sign language video.

So, as an additional input, we create a matrix $M \in R^{768\times n}$ where n is the number of frames and 768 the size of the BERT feature. Then, we fuse all the inputs : S_{prior}, M and the features related to the continuous signing video that contains the n frames by concatenating them. And then use the decoder that is already given to have better location of the signing subtitles in the continuous video.

We create M as follows: : at each time location of the spotting (of shape 768×1) in the matrix, we insert the average BERT feature of each of the corresponding words. And we complete the empty coordinates by 0. We use it as a pre-training strategy in order to obtain timings of the word annotations and to initialise the model weights and incorporate this knowledge through a single sign spotting task. Then, the model is trained to spot the single sign occurrence in a video window. Thus, we should have better results with this new model. (see other improvement ideas in [2] and [5])

7. Conclusion

To sum up our work, we would say that we have succeeded in understanding a current problem from a paper and a code already made and ready to be run: tasks regarding 'Aligning text to sign language video'. We tried to train a model on a subset of the dataset. We managed to get past the storage concerns and limitations of our computers and run the code to get results, despite the rather long running

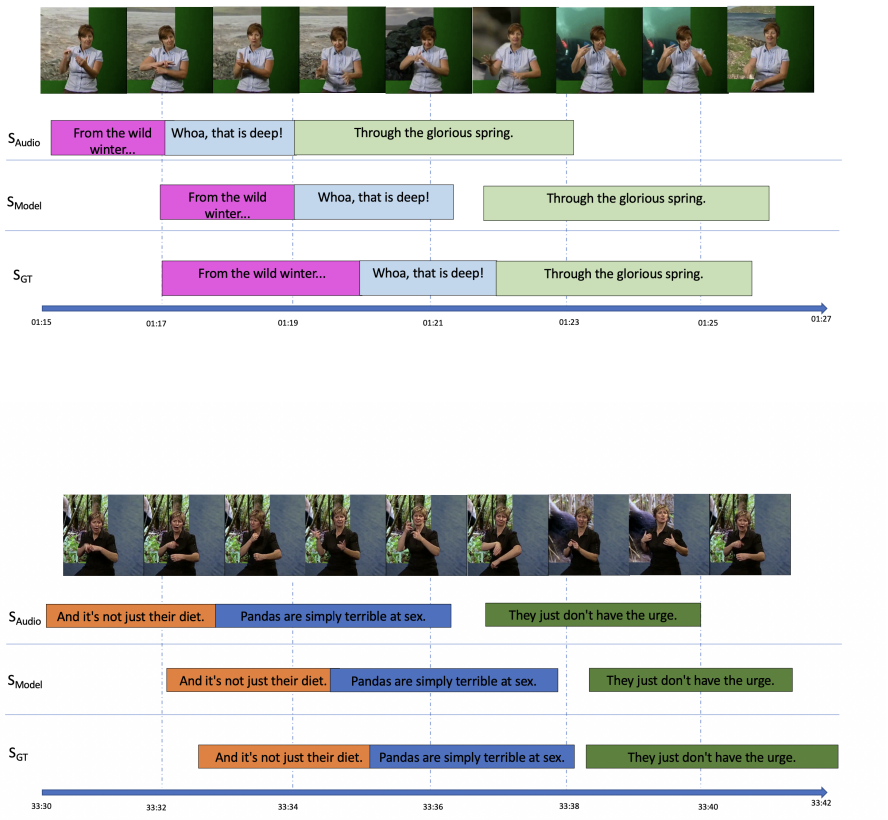

Figure 3. Visualizations of the text alignment predicted by our SAT model

time. Finally, we would say that the results obtained were quite satisfactory, especially because we only trained our model on a subset of our data and we obtained good scores. Finally, we would have liked to conclude with some important improvements, but we ran out of time to implement our idea.

References

[1] Samuel Albanie, Gül Varol, Liliane Momeni, Hannah Bull, Triantafyllos Afouras, Himel Chowdhury, Neil Fox, Bencie Woll, Rob Cooper, Andrew McParland, and Andrew Zisserman. BOBSL: BBC-Oxford British Sign Language Dataset. 2021.

[2] Afouras T. Varol G. Albanie S. Momeni L. Zisserman A. Bull, H. Aligning subtitles in sign language videos. 2021. In ICCV 2021.

[3] Hannah Bull, Triantafyllos Afouras, Gül Varol, Samuel Albanie, Liliane Momeni, and Andrew Zisserman. Aligning subtitles in sign language videos. In *ICCV*, 2021.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2017.

[5] K R Prajwal Samuel Albanie Gül Varol Andrew Zisserman Liliane Momeni, Hannah Bull. Automatic dense annotation of large-vocabulary sign language videos. 2022. In ECCV 2022.