# Topic K - Aligning Text to Sign Language Video

January 16, 2023

**Sébastien Meyer**
École Normale Supérieure Paris-Saclay, France
sebastien.meyer@ip-paris.fr

## Abstract

*Sign language alignment designates the task of aligning sequence of sign language frames with their corresponding subtitles, both being independently available to the user. Since the ordering and duration are very different between audio subtitles and subtitles for sign language, the alignment of the latter cannot be easily deduced from audio alignment. Also, signing corresponds to a translation of the subtitles, and not to a transcription. We based our project on the Subtitle Aligner Transformer model, which puts together BERT token embeddings of the subtitle text, I3D video features of the sequence and a prior location from the shifted temporal location of corresponding audio subtitles in a Transformer-based architecture. On top of this model, we add new prior vectors based on already available timestamps of single-sign spottings. We provide evaluation scores to assess our implementation, as well as empirical observations for the task of sign language alignment. Our code is available on GitHub*[1] *and our experiments are reproducible by using the provided parameters.*

## Introduction

With the increasing development of deep learning models for machine translation of both written and spoken languages, one could hope that their counterparts for machine translation of sign language has also achieved groundbreaking results. However, models for machine translation of sign language remains far from human performance, as explained by Koller in [3]. Many of the studies conducted on sign languages only cover small vocabulary tasks, while there is a clear lack of large annotated datasets and corresponding models. Indeed, state-of-the-art results in natural language processing have been achieved thanks to the access to large datasets and models.

The *Subtitle Aligner Transformer* (*SAT*) model introduced by Bull, Afouras, Varol, Albanie, Momeni and Zisserman in [2] addresses this issue by proposing a new Transformer-based architecture matching sequences of frames and corresponding signing subtitles, a task known as sign language alignment. Therefore, this model allows to annotate short sequences of sign language videos with their corresponding subtitles, in an automated fashion. Also, such an automated process paves the way for bridging gaps between the deaf and hearing people, by enhancing translation and language learning. In this project, we focused on the potential improvements we could make to the *SAT* model. To that extent, we append new priors about the temporal location of subtitles during the training phase of *SAT*. Already available word annotations of the video sequences under scrutiny can thus be added during the learning process and we can expect better scores at detecting subtitles' corresponding frames.

## 1. Method

In this section, we first describe the main idea behind our implementation. Then, we elaborate on the different prior vectors we tested. In particular, the first prior is the anchors prior, denoted $\mathbf{S}_{anchors}$, and the second prior is the spottings prior, denoted $\mathbf{S}_{spottings}$. Spottings prior appears in different forms, as we tried to optimize its impact on the predictions of our model, denoted *SAT-bis*.

The method behind our model aims at using the already annotated signs within the sequence during training and finetuning. Indeed, imagine that we possess a model trained to spot single signs in the video, either a previous model as in [5] for D or from the current *SAT* model as in [4] for D*. Based on this model, we know some prior information about where words from the subtitle might appear in the sign language video. Therefore, we are going to build different priors which can be concatenated to the video features and audio prior $\mathbf{S}_{prior}$ already of length $T$. As shown in **Figure 1**, these priors would be added to the base *SAT* architecture, only during training and finetuning. In **Section**

[1] https://github.com/sebastienmeyer2/subtitle_align

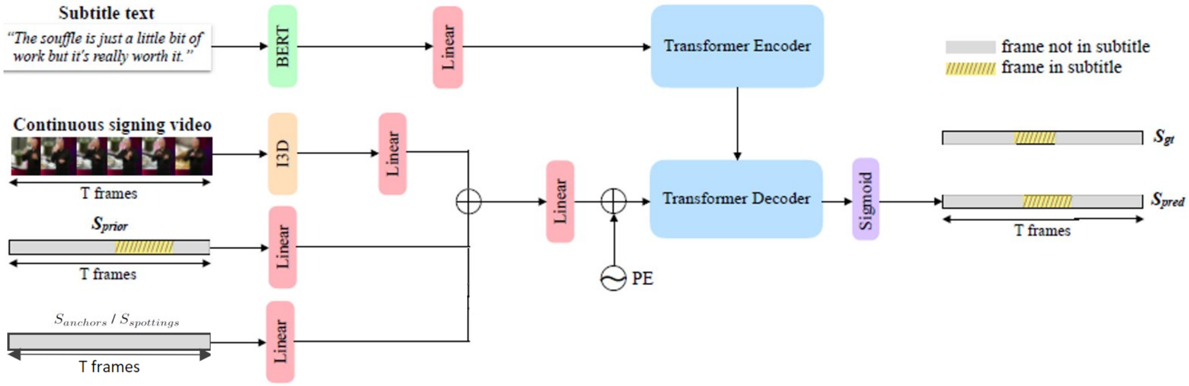

Figure 1. ***SAT-bis*** **model overview:** For the initial *SAT* model overview, see [2]. On top of the *SAT* architecture, we concatenate (only during training and finetuning) a new prior vector, either $\mathbf{S}_{anchors}$, $\mathbf{S}_{spottings}$ or both (see bottom left). This new prior is based on available word annotations.

2, we explain why we do not use these priors during word pretraining.

### 1.1. Anchors prior

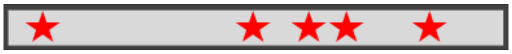

Figure 2. Anchors prior with M* and D* word annotations. Stars correspond to non-zero values in the vector.

The anchors prior $\mathbf{S}_{anchors}$, as shown in **Figure 2**, is a vector of length $T$ which contains zeroes in all frames, except those frames where words from the subtitle text have corresponding annotations in M* or D*. In that case, we use the probabilities output by the model as the values for the frames.

### 1.2. Spottings prior

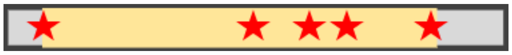

Figure 3. Spottings prior with M* and D* word annotations. Stars correspond to word annotations. Values are set to ones along the whole yellow line.

If we perform some sanity checks of the anchors prior, we can easily see during testing that only 0.66% of frames have non-zero values, opposed to a ground-truth proportion of 21.14% non-zero frames. Clearly, a model cannot learn from so sparse vectors. Instead, we are going to build a new spottings prior $\mathbf{S}_{spottings}$ based on the idea of the audio prior $\mathbf{S}_{prior}$. The range of frames between the first annotation and the last annotation belonging to the subtitle text will be set to ones, as shown in **Figure 3**, thus creating a new prior.

### 1.3. Adjusted spottings prior

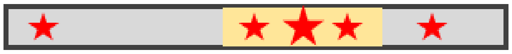

Figure 4. Adjusted spottings prior with M* and D* word annotations. Stars correspond to word annotations. Values are set to ones along the whole yellow line.

Again, a study of $\mathbf{S}_{spottings}$ reveals that the width of the prior is much larger to the one of ground-truth. During testing, we observe that the average width of $\mathbf{S}_{spottings}$ vectors is of 33.46 while the average width of $\mathbf{S}_{gt}$ vectors is of 26.43 and the average width of $\mathbf{S}_{prior}$ vectors is of 24.11. This stems from the fact that we detect annotations which might also belong to other subtitle texts. Therefore, we are going to use $\mathbf{S}_{prior}$ as a mean to adjust the width of $\mathbf{S}_{spottings}$. First, we select the median index of annotations, this will be the center of our new prior vector. Then, we create a range of ones with width of $\mathbf{S}_{prior}$ and center it around the median index. As shown in **Figure 4**, the width of the adjusted spottings vector $\mathbf{S}^{adj}_{spottings}$ is now much more smaller.

### 1.4. Adjusted spottings prior with PEN word annotations

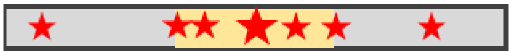

Figure 5. Adjusted spottings prior with M*, D*, P, E and N word annotations. Stars correspond to word annotations. Values are set to ones along the whole yellow line.

Finally, a last sanity check of $\mathbf{S}^{adj}_{spottings}$ shows that the majority of priors are simply empty vectors. Indeed, when using only the word annotations from M* and D*, the proportion of empty $\mathbf{S}^{adj}_{spottings}$ vectors is of 54.39%. In order to provide more non-empty priors, we will thus also use the P, E and N word annotations from [4]. These annotations come from different models: P is built by using a pretrained model for pseudo-labelling of the whole sequence, E is built by spotting small signing video clips already detected by previous models, and finally N is built by extending E with novel signs. By doing so, we define $\mathbf{S}^{adj+\text{PEN}}_{spottings}$ (see **Figure 5**), for which only 23.13% of the priors are empty vectors.

## 2. Experiments

In this section, we first describe the implementation details of our model. Then, we give some explanation about our data and evaluation metrics. We end by comparing our *SAT-bis* model to the baseline *SAT* model, in terms of evaluation metrics and examples of sign language alignment.

### 2.1. Implementation details

The main implementation details from [2] are kept, that is, the encoder and the decoder contain both two identical Transformers layers and heads of size $d_{model} = 512$. The input subtitle texts are transformed using a BERT model and the video features are extracted using a pretrained I3D model. Also, we use as $\mathbf{S}_{prior}$ the temporal location of the audio subtitles, shifted by 3.2 seconds in order to take into account the difference between audio and sign temporal locations.

As in [2], the training procedure is divided into three phases. The first phase, *word pretraining*, consists in training the model for the task of single-sign spotting, which is much easier than the task of sign subtitle alignment. Secondly, the model is trained on the task of sign language alignment, however the audio prior $\mathbf{S}_{prior}$ has only been manually annotated for a few dozen videos, so in order to train on a larger dataset, we also use the audio temporal location as the ground-truth temporal location. Finally, the model is finetuned on the task of sign subtitle alignment with the remaining videos for which we possess manual annotations of ground-truth temporal location of subtitles.

On top of the *SAT* architecture, we modify the linear layer applied before the Decoder (and before positional encoding). Indeed, we allow options to concatenate either $\mathbf{S}_{anchors}$, $\mathbf{S}_{spottings}$ or both during training. In such case, we add one or two linear layers of output size $d_{model}$ to embed the anchors and spottings priors. Then, the reprojection layer is increased of a size of one or two times $d_{model}$.

In our implementation, we replace the reprojection layer with fresh weights between pretraining and training. This is due to the fact that we do not use the anchors nor spottings priors during pretraining. Indeed, as during pretraining, subtitle text is equal to a single word, the anchors or spottings vector would bring enough information to the model for it to overfit. Another solution would be to initialize a complete reprojection layer, however keeping the part corresponding to the anchors or spottings priors frozen or fed with noise during word pretraining. In practice, it takes a few epochs to learn back the weights of the initial reprojection layer.

### 2.2. Data and evaluation metrics

In order to train our *SAT-bis* model, we use the BOBSL dataset from [1]. Compared to the initial models and results shown in [2] and [1], we slightly changed the training procedure. Indeed, we perform word pretraining using only 60 training and 10 validation videos picked at random, for 40 epochs. Then, training is performed on these same videos, during 40 epochs. Finally, finetuning is performed on all videos, however we changed the validation videos to be 4 different videos than those used in the base repository. Finetuning is performed during 50 epochs.

The final model is tested on the same set of videos as in [2] and [1], though its performance is expected to be poorer. We consider two main evaluation metrics: (i) frame-level accuracy, and (ii) $F1$-score. For the $F1$-score, hits and misses of subtitle alignment to sign language video are counted under three temporal overlap thresholds (IoU $\in 0.1$, 0.25, 0.50) between predicted $\mathbf{S}_{pred}$ and manually aligned $\mathbf{S}_{gt}$ subtitles, denoted as $F1@.10$, $F1@.25$, $F1@.50$, respectively.

### 2.3. Comparison to baselines

| Method | frame-acc | F1@.10 | F1@.25 | F1@.50 |
|---|---|---|---|---|
| *SAT* | 69.02 | **72.23** | **63.83** | **50.27** |
| $\mathbf{S}_{anchors}$ | **69.46** | 70.51 | 62.36 | 49.58 |
| $\mathbf{S}_{spottings}$ | 68.82 | 68.99 | 60.61 | 47.35 |
| $\mathbf{S}^{adj}_{spottings}$ | 69.18 | 69.29 | 61.26 | 48.51 |
| *SAT*$^{\text{PEN}}$ | 69.59 | 73.47 | 65.74 | 51.55 |
| $\mathbf{S}^{adj+\text{PEN}}_{spottings}$ | 69.03 | 70.60 | 62.57 | 49.51 |

Table 1. Results for different priors. Best method is in **bold**.

As a baseline, we use a model pretrained, trained and finetuned using the same videos and number of epochs, however without $\mathbf{S}_{anchors}$ and $\mathbf{S}_{spottings}$. This model achieves a frame accuracy of 69.02 and a $F1@0.50$ score of 50.27. First, we can compare the *SAT-bis* model when using $\mathbf{S}_{anchors}$ prior. Adding the anchors to the model increases the frame accuracy to 69.46. It seems that the model is able to spot some more specific frames for corresponding spottings, in particular since they are annotated. However, the overall $F1$ scores are worse when using $\mathbf{S}_{anchors}$ prior. The overall coverage of the ground-truth subtitle sequence is therefore less precise.

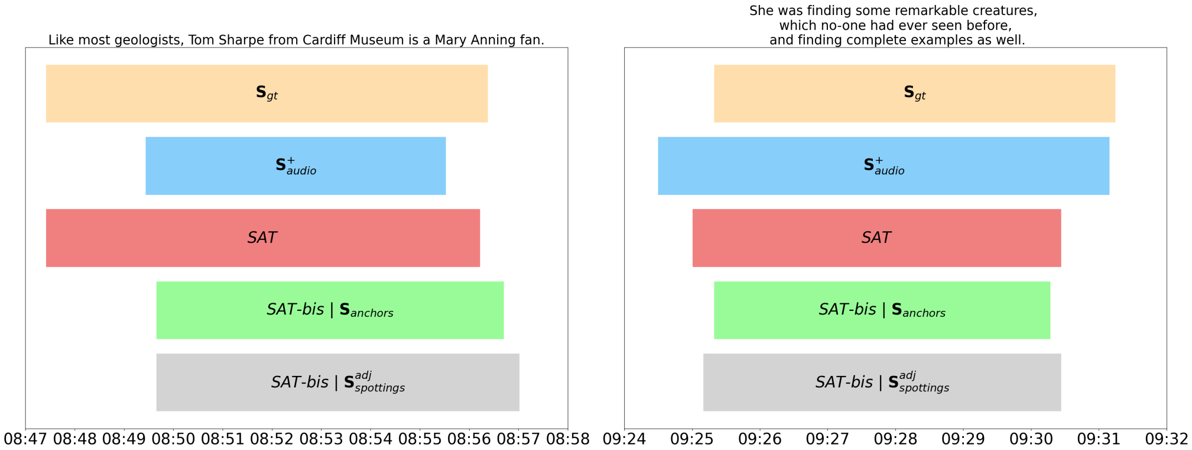

Figure 6. Two examples of sign language alignment selected during testing for comparing the base *SAT* model with our *SAT-bis* model, either with $\mathbf{S}_{anchors}$ or $\mathbf{S}^{adj}_{spottings}$.

Then, we can also compare the *SAT-bis* model when using $\mathbf{S}_{spottings}$. The initial prior vector, as discuss in **Section 1**, has a very large width and might include annotations from other subtitles. As we can see in **Table 1**, the results are worse than the Baseline *SAT* model. More specifically, the frame accuracy is quite similar to the baseline, however the $F1@.50$ went down from 50.27 to 47.35. Here come our improvements of the base $\mathbf{S}_{spottings}$ model. Adjusting the width of the spottings prior to the one of the audio prior helps improving all evaluation metrics. In particular, the $F1@.50$ score reaches 48.51.

Finally, the model including more word annotations from P, E and N again improves the results. We reach a frame accuracy of 69.03, which is equal to the frame accuracy of the *SAT* model. Also, the $F1@.50$ score becomes 49.51, which is still slightly slower than the $F1@.50$ of the *SAT* model. In order to compare this model to the baseline in a fair manner, we ran the learning procedure for *SAT* while also including P, E and N, thus yielding *SAT*$^{\text{PEN}}$. Compared to *SAT*$^{\text{PEN}}$, the *SAT-bis* model when using $\mathbf{S}^{adj+\text{PEN}}_{spottings}$ still gives worse results overall.

### 2.4. Qualitative analysis

## Conclusion

All in all, we have illustrated our modifications of the *SAT* model on simple examples. However, the overall results of our anchors and spottings priors remain lower than the results of the baseline *SAT* model. The main observation is that our very localized priors help the model in detecting specific frames, thus increasing the frame accuracy, while not giving more information about the whole temporal location of the subtitles, therefore decreasing the $F1$ scores.

Further research could be conducted on *SAT-bis*. More specifically, instead of creating priors based on the spottings probabilities, one could design a new prior taking into account word embeddings. These could either be the word embeddings of annotations, as in our anchors prior, or the embedding of part of a sentence, bounded by our (possibly adjusted) spottings prior. This might be expected to give more information to the model than a window vector of zeroes and ones.

## 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. *BBC-Oxford British Sign Language Dataset.* November 2021.

[2] Hannah Bull, Triantafyllos Afouras, Gül Varol, Samuel Albanie, Liliane Momeni and Andrew Zisserman. *Aligning Subtitles in Sign Language Videos.* 2021.

[3] Oscar Koller. *Quantitative Survey of the State of the Art in Sign Language Recognition.* August 2020.

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

[5] Liliane Momeni, Gül Varol, Samuel Albanie, Triantafyllos Afouras and Andrew Zisserman. *Watch, read and lookup: learning to spot signs from multiple supervisors.* October 2020.