Detecting Temporal Boundaries in Sign Language Videos

Jean QUENTIN
CentraleSupélec
3 Rue Joliot Curie, 91190 Gif-sur-Yvette
jean.quentin@student.ecp.fr

March 24, 2023

Abstract

Temporally segmenting signs in continuous sign videos is an important challenge that could dramatically lower the cost of annotating sign videos for computer vision models. Building upon the latest advances in action segmentation we propose an implementation that outperforms the current state of the art on the BSLCorpus dataset, by leveraging I3D video representations and a transformer-based classifier.

1. Introduction

While computer vision has seen many breakthroughs in the past few years, automatic sign language translation is still lagging behind in terms of performance. One of the main reasons is the lack of available annotated datasets of sign language videos rather than the technical challenge that translating signs represents. The time needed to produce the annotations is the key limitation that prohibits us from producing enough data to train regular computer vision algorithms on this task.

One way to reduce this cost is to pre-segment the signs in longer videos, so that annotators only have to label each segmented sign, which saves them a considerable amount of time.

Our work thus focuses on building upon the current advances in sign segmentation, and more generally in action segmentation and deep learning, to improve the current state of the art.

2. Related Work

Sign Segmentation : Renz *et al.* [11] have developed the current state of the art in sign language segmentation, they have significantly improved the performance compared to the previous methods [5]. When running the available code, we obtained a slightly different performance than the one stated in the paper, but which falls within the provided confidence intervals. Those results are available in Table 1. Their architecture is based on an I3D [1] backbone to extract features from videos and a multi-stage temporal convolutional network (MS-TCN) as proposed by Farha and Gall [6].

Method	mF1b	mF1s
I3D + MS-TCN (published)	68.68±0.6	47.71±0.8
I3D + MS-TCN (our results)	69.12	47.51

Table 1. Results obtained when running the code available on Renz *et al.*'s [11] GitHub page

Action Segmentation : While Renz *et al.* [11] focused on the I3D + MSTCN method [6], ASFormer [15] has been released since and exhibits very interesting performance. It leverages the Transformer architecture that has demonstrated to be very successful in many NLP and computer vision tasks [3, 2, 14]. Their model outperformed the state of the art on the action segmentation task on the 50 Salads dataset [13], a reference dataset that is composed of videos of people making salads.

3. Sign segmentation

3.1. Problem definition and evaluation metrics

We formulate the problem as stated by Renz *et al.* [11]: given a sequence $\mathbf{x} = (x_1, ..., x_n)$ of video frames, the goal is to predict the correct segmentation vector $\mathbf{y} = (y_1, ..., y_n) \in \{0, 1\}^{\mathbb{N}}$ where 1 is a boundary between two signs. The evaluation metrics are also the same, we will use mF1b, denoting the distance of a predicted boundary to the actual boundary, and mF1s which represents the IoU score of our segmentations compared to the ground truth. Since it

d_{hid} \ n_{heads}	2	4	8	16
200	47.73	47.82	**48.80**	45.00
400	45.80	47.62	45.54	47.02

Table 2. Results using mF1b score of the Transformer encoder only architecture

Method	mF1b	mF1s
Geometric features + RF (ref)	50.49	37.46
I3D + MS-TCN (ref)	68.68	47.71
I3D + ASFormer (untuned)	**69.47**	**49.31**

Table 3. Comparison of untuned ASFormer to previous state of the art methods

is hard to optimize a problem with two objective functions, we decided to rank solutions based on the mF1b, thus focusing on predicting the right boundaries, but we also displayed the mF1s to be consistent with Renz *et al.* [11].

3.2. Implementations

Since its release, the transformer architecture has enabled many breakthoughs in performance in NLP[2, 14] and in Computer Vision [3]. The goal of this project was thus to assess if we could improve the results of the original sign segmentation paper using a Transformer-based approach. We have thus proceeded to implement several architectures in order to review them and try to improve on the baseline of Table 1.

These implementations build upon the training pipeline developed by Renz *et al.* and we essentially try to improve the segmentation performance by changing the classifier. All the implementations and experiments can be found on my GitHub: jeanq1/sign-segmentation.

3.2.1 Vanilla transformer

We first tried to implement a vanilla transformer as described by Vaswani and al., and based on Harvard NLP group's Annotated Transformer [8], and upon not obtaining any decent results we implemented it with Pytorch's Transformer encoder and decoder layers [10] but we still didn't obtain anything decent. The models demonstrated very decent training performance but a very bad validation mF1b. This might be due to the poor decoding strategy we used (the greedy strategy), or to the relatively small dataset we used as suggested by [15].

3.2.2 Transformer encoder only

We then tried to implement a simpler architecture based on a transformer encoder with a fully connected layer as the decoder. It yielded better results (see Table 2) than the previous implementations but it still very far behind the other baselines.

3.2.3 ASFormer

ASFormer is a transformer-based architecture that was specifically designed to perform action segmentation on relatively small datasets (smaller than NLP datasets). What is even more interesting is that it was evaluated on the 50 Salads [13] dataset which is about the same size (in terms of frames) as the BSLCorpus [12] dataset and has similar content (videos focused on the hands movements, see Appendix A). And since ASFormer exhibited very good performance on the 50 Salads dataset, we decided to adapt their architecture, available on their GitHub (ChinaYi/ASFormer), to the sign segmentation task.
Their architecture is based on a Transformer encoder and three Transformer decoders (see Appendix B). The encoder outputs an embedding and predictions that are then passed on to the decoders which iteratively refine the predictions. They also make three major modifications to the original Transformer implementation: they add additional temporal convolutions to help constraint the search space of the model, they force lower-level attention layers to focus on local features and then gradually enlarge their footprints, and finally they modified the decoder architecture to allow it to attend to all positions in the refinement process.
The first implementation of the ASFormer yielded very promising results, since we only used the default configuration and it still outperformed the MS-TCN on both mF1b and mF1s.

4. Experiments

This part will focus only on experiments we have done with the ASFormer model, using the default attention mechanism. They have been performed on a K80 GPU on a GCP virtual machine.

4.1. Datasets

All experiments, as well as the first results presented in the last section are performed on the BSLCorpus [12] dataset preprocessed by Renz *et al.*, so as to keep only the 72k annotations that provide the sign category and temporal boundaries.

4.2. Impact of several hyperparameters

4.2.1 Impact of feature dimension

We first evaluated the impact of the feature dimension on the performance, in order to assess its importance. We dis-

Feature maps	mF1b	mF1s
16	68.60	46.61
32	**70.14**	49.00
64	69.47	**49.31**

Table 4. Study of the impact of the number of feature maps on the results

Number of decoders	mF1b	mF1s
1	68.89	48.52
2	68.92	**50.70**
3	**70.14**	49.00
4	69.87	49.48

Table 5. Study of the impact of the number of decoders on the segmentation performance (done with a feature dimension of 32)

covered that the feature dimension could be reduced from 64 to 32 to improve the quality of the predictions on the validation set and it also considerably reduced the training time. We estimate that this is due to the small size of our dataset and that lowering the embedding size can help the model overfit less on the training data. Table 4 displays the results we obtained when performing this experiment.

4.2.2 Impact of the number of decoders

We reproduced the experiment done in the ASFormer paper [15] in order to try to find an architecture that is better suited to our dataset. The results shown in Table 5 show that a three-decoder architecture is also the right choice for maximum performance on our dataset.

4.3. Experiments with loss functions

Since both Renz *et al.* and Yi *et al.* kept the same loss as in the original MS-TCN paper [6], we estimated that it would be interesting to see if we could find a loss function that would perform better on our dataset with the ASFormer architecture.
Thus, we have implemented the losses mentioned in the MS-TCN paper, namely the Cross-Entropy loss (CE), the Cross-Entropy loss with an additional smoothing loss (CE / T-MSE) and the Cross-Entropy loss with a Kullback-Leibler loss term (CE / KL).
Since there is class imbalance in our classification task (about 10% of frames are boundaries) we decided to implement a weighted Focal loss [9], which was designed to improve classification performance in this case. We used the Kornia [4] implementation to perform our computations.

We see in Table 6 that the Focal loss shows promising results compared to the original CE/T-MSE loss. We thus decided to evaluate the impact of the gamma parameter of

Loss Function	mF1b	mF1s
Cross Entropy	69.51	47.27
CE / T –MSE (original)	**70.14**	49.00
CE / LKL	69.32	50.05
Weighted Focal Loss (γ=2.0)	69.92	**50.24**

Table 6. Study of the impact of several loss functions on the results

Focal Loss Parameter γ	mF1b	mF1s
0.5	68.57	49.25
1.0	69.11	47.21
2.0	69.92	**50.24**
3.0	**70.61**	49.10
5.0	69.41	48.33

Table 7. Study of the impact of Focal Loss γ parameter on performance

Loss Function	λ	τ	mF1b	mF1s
CE / T –MSE	0.05	4.0	69.26	48.78
CE / T –MSE	0.15	4.0	**70.14**	49.00
CE / T –MSE	0.25	4.0	68.44	**50.27**
CE / T –MSE	0.15	3.0	69.17	48.44
CE / T –MSE	0.15	4.0	**70.14**	49.00
CE / T –MSE	0.15	5.0	68.28	**50.21**

Table 8. Study of the impact of λ and τ of the CE / T-MSE loss on classification performance

the Focal loss (see Table 7) and managed to outperform our previous best result of 70.14 mF1b with a 70.61 mF1b.
To be consistent in our experiments, we also tried to improve the classification performance of the CE/T-MSE loss by tuning the λ (importance of T-MSE compared to CE) and τ (truncation parameter of T-MSE, see [6]) parameters. The results are displayed in table 8, and show no improvement.

5. Conclusion

Our study has shown that naive implementations of transformer-based architectures perform quite poorly on our sign segmentation task. Instead, we demonstrated that using ASFormer, an architecture designed for action segmentation, enabled us to outperform the current state of the art in sign segmentation. Moreover, we showed that using the Focal loss, a loss designed for imbalanced datasets like ours, enabled us to obtain even better results. Future directions for this work include implementing the Hierarchical Action Segmentation Refiner proposed by Ahn and Lee [7], which can refine MS-TCN and ASFormer classification results.

Appendices

Appendix A. Comparison between 50 Salads dataset and BSLCorpus

Figure 1. Image taken from the 50 Salads dataset [13]

Figure 2. Image taken from the BSLCorpus dataset [12]

Appendix B. ASFormer architecture

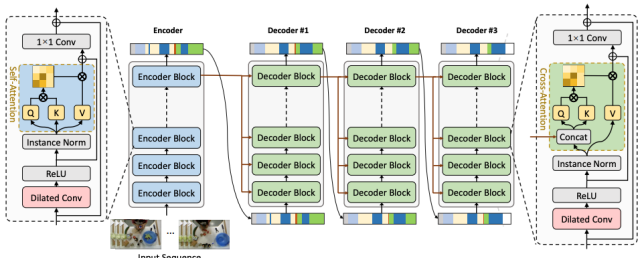

Figure 3. ASFormer model architecture

References

[1] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[4] D. Ponsa E. Rublee E. Riba, D. Mishkin and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020.

[5] Iva Farag and Heike Brock. Learning motion disfluencies for automatic sign language segmentation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7360–7364, 2019.

[6] Yazan Abu Farha and Juergen Gall. MS-TCN: multi-stage temporal convolutional network for action segmentation. *CoRR*, abs/1903.01945, 2019.

[7] Reza Ghoddoosian, Saif Iftekar Sayed, and Vassilis Athitsos. Hierarchical modeling for task recognition and action segmentation in weakly-labeled instructional videos. *CoRR*, abs/2110.05697, 2021.

[8] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017.

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[11] Katrin Renz, Nicolaj C. Stache, Neil Fox, Gül Varol, and Samuel Albanie. Sign segmentation with changepoint-modulated pseudo-labelling. *CoRR*, abs/2104.13817, 2021.

[12] Adam Schembri, Jordan Fenlon, Ramas Rentelis, Sally Reynolds, and Kearsy Cormier. British sign language corpus project: A corpus of digital video data and annotations of british sign language. *University College London*, 2017.

[13] Sebastian Stein and Stephen J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, page 729–738, New York, NY, USA, 2013. Association for Computing Machinery.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[15] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. *CoRR*, abs/2110.08568, 2021.