Computational Linguistics in the Netherlands Journal 12 (2022) 7-17 Submitted 12/2022; Published 12/2022

BeCoS Corpus: Belgian Covid-19 Sign Language Corpus. A Corpus for Training Sign Language Recognition and Translation

Vincent Vandeghinste[*†] VINCENT.VANDEGHINSTE@IVDNT.ORG
Bob Van Dyck[**] BOB.VANDYCK@KULEUVEN.BE
Mathieu De Coster[‡] MATHIEU.DECOSTER@UGENT.BE
Maud Goddefroy[†] MAUD.GODDEFROY@KULEUVEN.BE
Joni Dambre[‡] JONI.DAMBRE@UGENT.BE

[*] *Instituut voor de Nederlandse Taal, Leiden, the Netherlands*

[†] *Centre for Computational Linguistics, KU Leuven*

[**] *ESAT-PSI, KU Leuven*

[‡] *IDLab-AIRO – Ghent University – imec*

Abstract

We are presenting the Belgian Federal COVID-19 corpus, nicknamed the BeCoS (Belgian Covid Sign language) corpus. It consists of the entire archive of official press conferences from the Belgian Federal Government concerning the COVID-19 pandemic. The speakers speak mostly in Dutch or French and occasionally in German, and nearly all speech is accompanied by a deaf signer who performs live interpreting from what is being said. We have preprocessed the corpus with speaker diarisation, applied Belgian Dutch ASR, and post-ASR language identification and punctuation prediction as well as signer diarisation, sign language identification and sign language keypoint recognition. The corpus is made publicly available.

1. Introduction

Bridging the communicative gap between the deaf, the hard of hearing, and the hearing by providing automatic translation services between sign languages and spoken languages is the main goal of the SignON project.[1] Building such *sign language translation (SLT)* engines is a very challenging task, for a number of reasons, related to the lack of high quality parallel training data for SLT.

A first reason is the fact that sign languages usually do not come in writing, i.e., there is no standardized written form for sign languages which is used commonly within sign language communities, and machine translation (MT) systems usually suppose written language as their training data. It is therefore required to apply computer vision techniques to turn sign language videos into a form usable as source language in machine translation.

A second reason is the fact that there are not many freely available parallel datasets for sign languages. Flemish Sign Language (VGT – Vlaamse GebarenTaal) in combination with Dutch is a very low resource language pair, and the available data do not provide enough food for the current data hungry neural MT approaches.

A third reason is the fact that for most parallel datasets the sign language is a translation or interpretation of the spoken language. It is therefore a form of *translationese* sign language, as the spoken language is the source and the sign language is the target of the original translation/interpretation.

A fourth reason is the fact that the signer needs to sign *authentic* sign language. In the case of interpreting of spoken languages, it is normal professional practice that the interpreter is a native

1. `https://signon-project.eu/`

©2022 Vincent Vandeghinste, Bob Van Dyck, Mathieu De Coster, Maud Goddefroy, Joni Dambre.

speaker of the target language (just as is the case with translators). This, of course, poses difficulties in the case of interpretation into sign languages, as the interpreters have to hear the source they have to interpret. Therefore, the interpreter is most often not a first-language (L1) signer and not part of the *authentic* sign language community.

During the COVID-pandemic (2020 - 2022?) sign languages suddenly became very visible to the general public, as the governmental press conferences, aimed at all citizens, were live and in real-time interpreted into sign languages. This was one of the triggers to collect these data and use them within the SignON project.

With the dataset described in this paper we address most of the aforementioned reasons. The first reason (lack of a standardized written form) is partially addressed by providing automatically extracted keypoints from the sign language interpreters, providing the corpus user with a first, admittedly still very raw, reduction of the signer's data compared to the video signal. The second reason (scarcity of parallel datasets) is addressed by making this corpus freely available for further research. The third reason (sign language as a target language) is not addressed as it still concerns an interpretation from spoken Dutch into VGT. The fourth reason (the need for authentic signing), though, maybe surprisingly, is addressed. The SL interpreters are members of the deaf community. The spoken language utterances are first interpreted by a hearing SL interpreter, and this signal is re-interpreted by a deaf SL interpreter, resulting in *authentic* L1 SL. Whether this chain of interpretation leads to information loss or change would be a topic of study in its own right, but unfortunately we could not get hold of the videos of the intermediate interpretation.

Section 2 describes where to find resources for SLT. Section 3 describes the BeCoS corpus and how it was processed. Section 4 draws some conclusions, sketches possible future work and describes the availability of the corpus for researchers.

2. Related work

There are several, albeit rather small, datasets that contain sign language. A recent overview can be found in Kopf et al. (2021) and a compendium containing updated information was presented in Kopf et al. (2022). It has to be noted, though, that many of these data resources are not easily accessible and downloadable, let alone suitable, for NLP research (De Sisto et al. 2022).

For VGT, there is the Corpus Vlaamse Gebarentaal (Van Herreweghe et al. 2015). This corpus has the advantage of being an 'SL as the source'-corpus. It is largely built from elicited material, and was built for the purpose of linguistic study of VGT, not for machine learning. It is rather small according to NLP standards. It is also not (yet) fully translated nor annotated, and is therefore of limited use in the context of SLT. This corpus (currently) contains 2737 parallel sentences between glosses and Dutch text, which is insufficient for SLT systems. We have made this corpus available for download at `http://hdl.handle.net/10032/tm-a2-v6`.

Another VGT dataset, explicitly geared towards machine learning is the Content4All dataset (Camgöz et al. 2021). This dataset consists of television broadcasts, covering the domains *news* and *weather*, associated with sign language interpretation. Downsides of these data are that the SL is the target language, and the interpreters are hearing interpreters. An upside in the context of SLT is that pose information of the signers, which was automatically extracted with OpenPose (Cao et al. 2019), is made available for download. Another plus is that the closed caption subtitles are also made available, providing a written representation of what is being said, leading to a truly parallel corpus.

Within the SignON project we are collecting more resources containing VGT, such as the proceedings of the plenary sessions of the Belgian Federal Parliament, which have been interpreted for some time. Processing of these data in a similar way as the BeCoS dataset is ongoing and can be expected in the near future, but the data suffers from several drawbacks, such as VGT as a target, and hearing non-L1 interpreters.

We are also in the process of collecting VGT-as-a-source material, but this process is still too premature to provide more details.

3. The corpus

3.1 Dataset description

The videos that constitute the raw source material of this corpus have been downloaded from `https://news.belgium.be/nl/corona`, which contains the Belgian federal COVID-19 press conference streaming archive.

A total of 220 videos of press conferences are available, for a total running time of 7 days, 9 hours, 34 minutes, 23 seconds. All but seven videos contain sign language interpretation. There are multiple speakers per press conference.

The corpus is multilingual, in the sense that, as typically is the case at the federal level in Belgium, speakers may speak Dutch, French or (to a much lesser extent) German. A single speaker may switch language, leading to parts which are in non-native speech. When the original speech is in Dutch, it is interpreted in VGT; when the original speech is in French, it is interpreted in LSFB (Langue des Signes de Belgique Francophone – *Sign Language of Francophone Belgium*).

3.2 Preprocessing

Ideally, we could perform some processing on the audio, such as audio language identification, audio transcription for Belgian Dutch, audio transcription for Belgian French, speaker diarisation, and utterance segmentation. Section 3.2.1 presents the audio processing.

Concerning the sign language interpretation, ideally we could perform sign language recognition, signer diarisation and utterance segmentation. Sign language preprocessing is described in section 3.2.2.

3.2.1 Audio processing

We did not find any freely available tools that would allow us to perform language identification on a multilingual audio signal, indicating from when to when which language was used. The approach of Valk and Alumäe (2021) for language identification[2] classifies a wave file according to the language spoken, with an error rate of 6.7%. But in our case the data consist of long multilingual files, and building a language classifier based on Valk and Alumäe (2021) that classifies a sliding window of audio in order to determine the exact moments of code switching was beyond the scope of the current work. Future releases of the automated corpus annotations may use such an approach. For now, we had to come up with a work around for this problem.

For the audio transcription of Belgian Dutch, we could rely on Van Dyck et al. (2021), which is a Kaldi-based (Povey et al. 2011) ASR system, with an acoustic model for Belgian Dutch, trained on the Spoken Dutch Corpus (Oostdijk et al. 2002) and a general language model, trained on newspapers and Dutch corpora. It outputs files in the *ctm*[3] format which contains the recognized words, timestamps, and confidence values per word. This information is converted into ELAN annotation format as a separate tier. ELAN (Wittenburg et al. 2006) is a well known software environment for linguistic annotation, and is also often used in the sign language corpus community.

The result of the previous step is that we sometimes apply a Belgian Dutch ASR onto spoken French, which results in rather non-sensible Dutch output with relatively low confidence values. By filtering out low confidence word sequences we can determine with reasonable accuracy where French audio occurs.

2. As available on `https://huggingface.co/speechbrain/lang-id-voxlingua107-ecapa`
3. CTM stands for time-marked conversation file.

We used a sliding window $W_k = \{w_k, w_{k+1}, ..., w_{k+n}\}$ of ASR output words $W = \{w_1, ..., w_z\}$ of length $n = 5$, and a threshold $\theta = 0.9$. If within W_k there are more than $l = 3$ words that have a confidence value c where $c > \theta$, we consider the language to be Dutch. If only two words have a $c > \theta$, we check whether the words previous to the current W_k, i.e. w_{k-2} and w_{k-1} were classified as Dutch. Otherwise the language for the words in W_k was classified as non-Dutch. The values of these parameters were determined on a small development set. Determining the exact borders of these sequences proves to be more difficult. A formal evaluation on how well the filtering works, and whether a different threshold might yield better results remains to be done.

For Belgian French speech recognition, we did not find any freely available tools that could run on our hardware and deal with rather long audio sequences (often more than 1 hour). For future versions of the automated corpus annotations, we will test whether the approach of Grosman (2021) available at `https://huggingface.co/jonatasgrosman/wav2vec2-large-fr-voxpopuli-french` provides a viable solution.

For speaker diarisation we used a web service through the CLARIN Infrastructure at `https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/SpeakDiar` (Bredin et al. 2019) that takes audio as input and returns an ELAN annotation file with a speaker diarisation tier. We have not formally evaluated how well speaker diarisation works for Dutch, but spot checks showed good results.

Utterance segmentation is performed on the basis of the recognized speech, taking the diarisation into account, and is using the approach described in Vandeghinste and Guhr (2022). It is a transfer learing approach which finetuned the RobBERT transformer language model (Delobelle et al. 2020) onto a punctuation and segmentation classification task.

The total number of utterances, be it delimited through speaker diarisation, silence (threshold of two seconds), or segmentation prediction amounts to 52,799, totaling 671,326 recognized words that have been classified as Dutch. The outputs of each of the audio processing steps are available in separate ELAN tiers, and can be used for further research.

We have also extracted all text in txt files and processed them with the Frog linguistic analysis suite (Van den Bosch et al. 2007), making the data available in FoLiA (van Gompel and Reynaert 2013) and tab-separated form.

3.2.2 Sign Language Processing

Sign Language Identification The identification of which sign language is being signed is a challenging problem. It is compounded in the case of this corpus, as VGT and LSFB are related languages that share many similarities (Vermeerbergen et al. 2013). However, the persons interpreting VGT in the COVID-19 press conferences are different from those interpreting LSFB. Therefore, we can rely on signer diarisation to detect changes in the sign language.

Signer Diarisation The first stage in sign language processing is signer diarisation, i.e., determining when there is a signer change. In the current corpus, this usually indicates a change in sign language as well. Signer diarisation on this corpus comes with several challenges. The deaf interpreter is shown picture-in-picture in the bottom right of the press conference video: this box needs to be extracted. The image quality of this picture-in-picture video differs between videos: the resolution (including aspect ratio) and frame rate can differ. An additional difficulty is that the interpreters for VGT and LSFB walk in and out of frame when the spoken language changes.

We take a multi-step approach towards signer diarisation to account for these difficulties. First, we detect the coordinates of the picture-in-picture view: more specifically we employ a person detector and crop the video according to the resulting coordinates. Second, we automatically segment the video whenever the interpreter changes. We do this by comparing face embeddings in a latent space.

The picture-in-picture view of the interpreter is always in the bottom right. We use YOLOv5 (Jocher et al. 2021) to detect all people in one frame of the video. By applying YOLOv5 to a

frame of the video, we obtain a set of bounding boxes B. A bounding box $b \in B$ is a 4-tuple: $b = (x_{min}, y_{min}, x_{max}, y_{max})$. The interpreter bounding box b is determined as,

$$b_i = a\left(\underset{y_{min},x_{min}}{\text{argmax}}\ B\right), \tag{1}$$

that is, it is the bounding box with the highest values for the top left coordinate from among all bounding boxes in B. The function a in equation 1 adapts the bounding box to account for the interpreter moving around in the duration of the video. It sets the maximum x and y values (i.e., the bottom right coordinate) to the full pixel resolution of the video. It also decreases the top left coordinate by 10% of the full pixel resolution of the video. Let w and h be the (constant) width and height of the video. Then,

$$a(b) = \left(x_{min} - \frac{w}{10}, y_{min} - \frac{h}{10}, w, h\right). \tag{2}$$

This approach can be performed automatically on all videos in the corpus. The results are then visually inspected and corrected where necessary. In case no interpreter was present in the recording (and thus there is no SL information), this is spotted during the visual inspection stage, and such files are not processed further. This is the case for seven out of 219 files (3%).

We use FaceNet (Schroff et al. 2015) to detect the face of the interpreter and compute a face embedding. This allows us to effectively compare two face images by computing the distance between two corresponding vectors. Whenever the face embedding changes by a sufficient amount, we create a new segment. In order to account for small variations that may occur due to noise or due to the arms or hands of the interpreter obstructing their face, we use a simple moving average to smooth out the predictions.

We keep two buffers p (previous) and c (current). These buffers are equally long sequences of face embeddings. The buffer length M is equal to the amount of frames per second of the video: together, p and c account for two seconds in the video.

The buffers are filled in a first-in-first-out (FIFO) manner with a warm-up stage. In the warm-up stage, first p is filled with the face embeddings e of N video frames. Then, c is filled with the subsequent N face embeddings. If, for a given frame, no embedding is found, then that frame is skipped. After the warm-up period, p and c are rotated. Let

$$p^{(N)} = (e_1, e_2, \ldots, e_M) \tag{3}$$

and

$$c^{(N)} = (e_{M+1}, e_{M+2}, \ldots, c_N) \tag{4}$$

be the buffers after $N = 2M$ frames (e.g., after the warm-up period), then

$$p^{(N+1)} = (e_2, e_3, \ldots, e_M, e_{M+1}) \tag{5}$$

and

$$c^{(N+1)} = (e_{M+2}, e_{M+3}, \ldots, e_N, e_{N+1}) \tag{6}$$

are the buffers after $N + 1$ frames.

These buffers are used to implement the simple moving average of the Euclidean distance between face embeddings of frames that are M apart. We thus compute a new sequence of M distances $d = (d_1, d_2, \ldots, d_M)$, where $d_i = ||c_i - p_i||_2$. When the average Euclidean distance exceeds a threshold θ, we create a new segment. That is, when

$$\frac{1}{M}\sum_{i=1}^{M} d_i > \theta. \tag{7}$$

We set $\theta = 1$ (empirically found). At this point, the buffers p and c are flushed and a new warm-up period is triggered to avoid satisfying the condition in Eq. 7 multiple times in short succession.

As a result, we get automatic boundary detection whenever the face embedding changes. This means that we do not need to label the entire videos manually, but simply need to annotate individual segments as VGT or LSFB. This is performed with a custom-made tool. Due to false positive boundary detections, we may have two or more segments with identical language annotations. These segments are merged automatically in a post-processing step.

The entire process of signer diarisation is illustrated in figure 1.

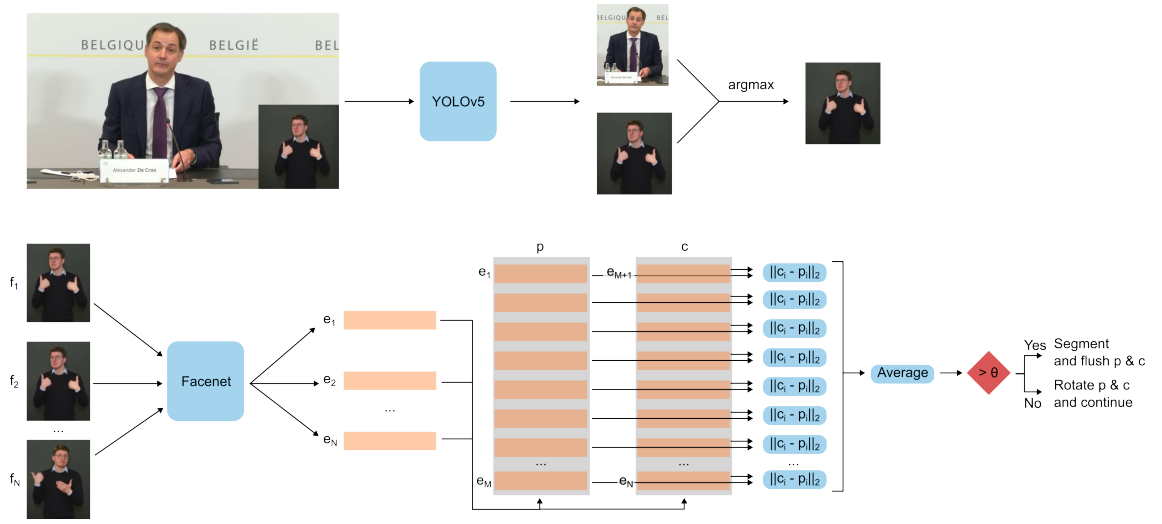

Figure 1: The signer diarisation process. Top: extracting the picture-in-picture view of the interpreter using YOLOv5. Bottom: detecting interpreter changes by comparing face embeddings obtained using Facenet.

Feature Extraction The second stage in sign language processing is providing a sign language representation that can readily be used for sign language *recognition* or *translation*. Similar to previous works, e.g., Camgöz et al. (2021), we provide human pose keypoints extracted from the video. Such keypoints can be used as input features for recognition and translation models without further processing by the user of the corpus. We use the MediaPipe toolkit (`https://google.github.io/mediapipe/`) because it is easy to install and use, has a low computational cost, and provides keypoints of sufficient quality to train recognition models with (Moryossef et al. 2021).

The MediaPipe toolkit extracts 75 keypoints per video frame, i.e. 33 body pose keypoints, see figure 2, and 21 hand keypoints per hand, see figure 3. The keypoints are normalized with respect to the image dimensions. Missing values (i.e., keypoints that MediaPipe could not predict) are indicated as not-a-number (NaN) values. These values can be replaced using an imputation approach in the SLT models that make use of this corpus.

These 75 keypoints, in three dimensions, amount to 225 datapoints per frame. The videos have framerates of 25 frames or 30 frames per second, so that amounts to 5625 or 6750 datapoints per second. It would be too much of a burden to put these datapoints on ELAN's shoulders, so these data is delivered in separate files.

These datapoints can form the starting point for sign language *recognition* or *translation*. Examples of such "pose based" models can be found in scientific literature, e.g., (Moryossef et al. 2021), (Camgöz et al. 2021), (Orbay and Akarun 2020). The conversion of an ELAN SL dataset to a dataset that is ready for use in machine translation models is described in (De Sisto et al. 2022).

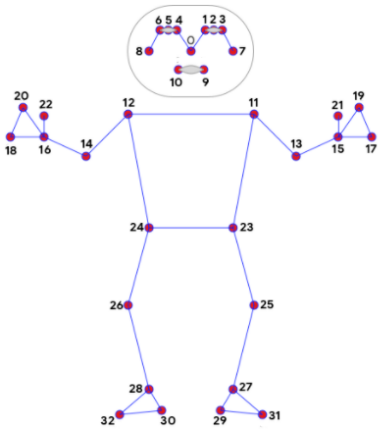

Figure 2: MediaPipe Body Keypoints. Source: `https://google.github.io/mediapipe/solutions/pose.html`. Accessed: September 12, 2022.

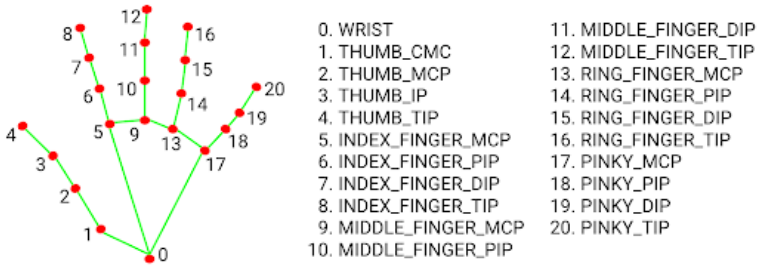

Figure 3: Mediapipe Hand Keypoints. Source: `https://google.github.io/mediapipe/solutions/hands.html`. Accessed: September 12, 2022.

3.2.3 Interaction between audio and sign language processing

Although one might think that when determining the time stamps of signer change, this information could be used to determine the time stamps of language change in the speaker, this does not seem to be the case. As it concerns unscripted live videos without post-editing, the signers have to follow the order of speech events, and always lag behind.

4. Conclusions and future work

4.1 Conclusions

We have collected the videos of all the official COVID-19 press conferences of the Belgian Federal government. These have been automatically annotated with Belgian Dutch speech recognition, post-ASR language identification, speaker diarisation and segmentation.

Nearly all of these videos contain live sign language interpretation by deaf interpreters. Spoken Dutch is interpreted into VGT, Spoken French is interpreted into LSFB. The signing signal has been automatically annotated with signer diarisation and keypoint recognition. All but the keypoint recognition have been integrated into a single ELAN file per press conference, with the different annotation layers into different tiers.

With this dataset we hope to partially lessen the lack of sufficient resources for building an SLT system from VGT to Dutch or the other way around. We are fully aware of the limitations of our automatic processing but wanted to release the dataset as it is now, as SLT research is in dire need of such data.

In order to use the dataset for training actual sign language translation from VGT into Dutch, we advise the approach and tools described in De Sisto et al. (2022), which homogenizes different formats sign language datasets come in into a format which can be and has been used for machine learning purposes.

4.2 Future work

While the processing of the dataset as described was already a significant effort, some aspects leave ample room for further improvement, and will hopefully be addressed in future versions of the corpus annotations.

The Belgian Dutch ASR system was taken as it came, without domain adaptation with respect to the language model or the lexicon. By training the ASR language model on domain specific data, and extracting a lexicon from such data, recognition results could be improved.

The post-ASR language identification can surely be improved, although we have not yet been able to formally evaluate the current approach. We expect that training a classifier that makes use of pretrained language models could improve the results. An approach that uses spoken language identification on the audio, as described in Valk and Alumäe (2021) and mentioned in section 3.2.1 or that combines both the post-ASR language identification with the spoken language identification would probably be even better. If this is the case, we will rerun language identification and subsequent processing steps in a next release of the automatic annotations.

We would also like to apply Belgian French ASR to the spoken Belgian French in order to also preprocess the Belgian French-LSFB parallel pair in a similar matter than the Belgian Dutch-VGT language pair. For future versions of the automated corpus annotations, we will test whether the approach of Grosman (2021)[4] provides a viable solution.

Another matter concerns the segmentation of sign language utterances. While there is research on sign segmentation, i.e., detecting the borders of individual signs (Renz et al. 2020, Renz et al. 2021, De Sisto et al. 2021), segmentation at the utterance level is rarely discussed (Mesch and Müller de Quadros 2019) and is not applied on these data.

4. Available at `https://huggingface.co/jonatasgrosman/wav2vec2-large-fr-voxpopuli-french`

Furthermore, the recognition of the SL keypoints still results in rather raw data. Subsequent annotations could interpret these keypoints into movement recognition or even lexicalized sign recognition, leading to automatically generated glosses.

Additional automatic annotation of the transcribed Dutch with part-of-speech tagging and lemmatization is also still on our to-do list.

Ideally, (part of) the dataset would be manually transcribed and annotated, to evaluate or even train new and better models, but this is out of the scope of the SignON project.

4.3 Availability of the dataset

The dataset is available for download for researchers. The persistent identifier of the landing page is `http://hdl.handle.net/10032/tm-a2-v5`.

Acknowledgements

A large part of the work in this paper is financed by the SignON project. This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101017255.

The use of Frog linguistic annotation is funded by CLARIAH-NL through the ClaSaBeD project.

Mathieu De Coster's research is funded by the Research Foundation Flanders (FWO Vlaanderen): file number 77410.

References

Bredin, Hervé, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill (2019), pyannote.audio: neural building blocks for speaker diarization. https://arxiv.org/abs/1911.01255.

Camgöz, Necati Cihan, Ben Saunders, Guillaume Rochette, Marco Giovanelli, Giacomo Inches, Robin Nachtrab-Ribback, and Richard Bowden (2021), Content4all open research sign language translation datasets, *CoRR*. https://arxiv.org/abs/2105.02351.

Cao, Z., G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh (2019), Openpose: Realtime multi-person 2d pose estimation using part affinity fields, *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

De Sisto, Mirella, Dimitar Shterionov, Irene Murtagh, Myriam Vermeerbergen, and Lorraine Leeson (2021), Defining meaningful units. challenges in sign segmentation and segment-meaning mapping (short paper), *Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*, Association for Machine Translation in the Americas, Virtual, pp. 98–103. https://aclanthology.org/2021.mtsummit-at4ssl.11.

De Sisto, Mirella, Vincent Vandeghinste, Santiago Egea Gómez, Mathieu De Coster, Dimitar Shterionov, and Horacio Saggion (2022), Challenges with sign language datasets for sign language recognition and translation, *Proceedings of the Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, pp. 2478–2487. https://aclanthology.org/2022.lrec-1.264.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2020), RobBERT: a Dutch RoBERTa-based Language Model, *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, pp. 3255–3265. https://aclanthology.org/2020.findings-emnlp.292.

Grosman, Jonatas (2021), Fine-tuned French Voxpopuli wav2vec2 large model for speech recognition in French.

Jocher, Glenn, Alex Stoken, Ayush Chaurasia, Jirka Borovec, NanoCode012, TaoXie, Yonghye Kwon, Kalen Michael, Liu Changyu, Jiacong Fang, Abhiram V, Laughing, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Jebastin Nadar, imyhxy, Lorenzo Mammana, AlexWang1900, Cristi Fati, Diego Montes, Jan Hajek, Laurentiu Diaconu, Mai Thanh Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106 (2021), ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support. https://doi.org/10.5281/zenodo.5563715.

Kopf, Maria, Marc Schulder, and Thomas Hanke (2021), Overview of datasets for the sign languages of Europe, *Deliverable 6.1*, Easier project.

Kopf, Maria, Marc Schulder, and Thomas Hanke (2022), The sign language dataset compendium: Creating an overview of digital linguistic resources, *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources*, European Language Resources Association, Marseille, France, pp. 102–109. https://aclanthology.org/2022.signlang-1.16.

Mesch, Johanna and Ronice Müller de Quadros (2019), Segmentation in sign languages, *TISLR13, the 13th conference of Theoretical Issues in Sign Language Research.*

Moryossef, Amit, Ioannis Tsochantaridis, Joe Dinn, Necati Cihan Camgoz, Richard Bowden, Tao Jiang, Annette Rios, Mathias Muller, and Sarah Ebling (2021), Evaluating the immediate applicability of pose estimation for sign language recognition, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3434–3440.

Oostdijk, Nelleke, Wim Goedertier, Frank van Eynde, Louis Boves, Jean-Pierre Martens, Michael Moortgat, and Harald Baayen (2002), Experiences from the spoken Dutch corpus project, *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, European Language Resources Association (ELRA), Las Palmas, Canary Islands - Spain. http://www.lrec-conf.org/proceedings/lrec2002/pdf/98.pdf.

Orbay, Alptekin and Lale Akarun (2020), Neural sign language translation by learning tokenization, *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, IEEE, pp. 222–228.

Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely (2011), The kaldi speech recognition toolkit, *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Renz, Katrin, Nicolaj C. Stache, Neil Fox, Gül Varol, and Samuel Albanie (2021), Sign segmentation with changepoint-modulated pseudo-labelling. https://arxiv.org/abs/2104.13817.

Renz, Katrin, Nicolaj C. Stache, Samuel Albanie, and Gül Varol (2020), Sign language segmentation with temporal convolutional networks, *CoRR*. https://arxiv.org/abs/2011.12986.

Schroff, Florian, Dmitry Kalenichenko, and James Philbin (2015), Facenet: A unified embedding for face recognition and clustering, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823.

Valk, Jörgen and Tanel Alumäe (2021), VoxLingua107: a dataset for spoken language recognition, *Proc. IEEE SLT Workshop.*

Van den Bosch, A., G.J. Busser, W. Daelemans, and S. Canisius (2007), An efficient memory-based morphosyntactic tagger and parser for Dutch, *in* van Eynde, F., P. Dirix, I. Schuurman, and V. Vandeghinste, editors, *Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting*, Centre for Computational Linguistics, Leuven, Belgium, pp. 99–114.

Van Dyck, Bob, Bagher BabaAli, and Dirk Van Compernolle (2021), A Hybrid ASR System for Southern Dutch, *Computational Linguistics in the Netherlands Journal* **11**, pp. 27–34. https://clinjournal.org/clinj/article/view/119.

van Gompel, Maarten and Martin Reynaert (2013), Folia: A practical xml format for linguistic annotation – a descriptive and comparative study, *Computational Linguistics in the Netherlands Journal* **3**, pp. 63–81. https://clinjournal.org/clinj/article/view/26.

Van Herreweghe, M., M. Vermeerbergen, E. Demey, H. De Durpel, and S. Verstraete (2015), Het Corpus VGT. Een digitaal open access corpus van videos and annotaties van Vlaamse Gebarentaal, ontwikkeld aan de Universiteit Gent i.s.m. KU Leuven. www.corpusvgt.be.

Vandeghinste, Vincent and Oliver Guhr (2022), FullStop: Punctuation and Segmentation Prediction for Dutch with Transformers, *Computational Linguistics in the Netherlands Journal*.

Vermeerbergen, Myriam, Jan Nijen Twilhaar, and Mieke Van Herreweghe (2013), Variation between and within Sign Language of the Netherlands and Flemish Sign Language, *Language and Space Volume 30 (3): Dutch*, De Gruyter Mouton, Berlin, pp. 680–699.

Wittenburg, Peter, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes (2006), ELAN: a professional framework for multimodality research, *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, European Language Resources Association (ELRA), Genoa, Italy. http://www.lrec-conf.org/proceedings/lrec2006/pdf/153_pdf.pdf.