APPENDIX
Automatic dense annotation of large-vocabulary sign language videos

Liliane Momeni[1⋆], Hannah Bull[2⋆], Prajwal K R[1⋆],
Samuel Albanie[3], Gül Varol[4], and Andrew Zisserman[1]

[1] Visual Geometry Group, University of Oxford, UK
[2] LISN, Univ Paris-Saclay, CNRS, France
[3] Department of Engineering, University of Cambridge, UK
[4] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, France
{liliane,prajwal,albanie,gul,az}@robots.ox.ac.uk;
hannah.bull@lisn.upsaclay.fr
https://www.robots.ox.ac.uk/~vgg/research/bsldensify/

We first provide a discussion on the dependency and complementarity between different annotation approaches (Sec. A). We subsequently describe implementation details in Sec. B, present additional experimental results in Sec. C and show some qualitative examples and failure cases in Sec. D.

A Different automatic annotation approaches

We provide a summary of the different approaches mentioned in this paper for annotating signs automatically in sign language interpreted TV shows, which consist of continuous signing and weakly-aligned English subtitles. We highlight specifically the limitations of different approaches and their dependencies.

- M refers to automatic sign annotations obtained in previous work [2] from mouthings, as signers often mouth a word and sign it simultaneously. Specifically, the sign annotations are obtained by querying subtitle words in a signing window with a mouthing-based keyword spotting model and saving the most confident model predictions. Mouthing is a strong signal, but it cannot be used to annotate all data (since signers do not mouth continuously). Furthermore, these automatic annotations are skewed towards words with 'easy' mouthings.
- D refers to automatic sign annotations obtained in previous work [6] by leveraging online sign language dictionary clips. In more detail, a joint embedding space is learned between the *isolated* dictionary video clips and the *continuous* signing video sequences. At inference time, the cosine similarity between the continuous signing sequence and dictionary clips corresponding to subtitle words is calculated. The sign annotations correspond to the dictionary clips with highest similarity. Although these automatic annotations are not limited to signs accompanied by mouthings, they are limited to the

⋆ Equal contribution

vocabulary of the online sign dictionary. Furthermore, they are biased to an extent towards mouthings since the joint embedding space is learned using M annotations.

- A refers to automatic sign annotations obtained in previous work [10] by using the localisation ability from the attention mechanism of a video-to-text Transformer model. The encoder takes as input pre-computed video features (from a sign recognition model trained with M and D annotations) and outputs a sequence of word stems. The sign annotations correspond to words which are correctly predicted and the sign timestamps are obtained by looking at the temporal position where the encoder-decoder attention is maximised. Compared to mouthing (M) and dictionary (D) annotations, the attention (A) annotations are obtained by taking context into account.

- M* refers to new and improved mouthing annotations obtained in this work. In fact, we upgrade to a state-of-the-art keyword spotting model (Transpotter [8]) and finetune this model on signer mouthings. We also use subtitles which are better aligned to the signing for centering our querying windows. This enables the number of detected mouthings and therefore automatic sign annotations to be greatly expanded.

- D* refers to new and improved dictionary annotations obtained in this work by (i) using subtitles which are better aligned to the signing for centering our querying windows, and (ii) expanding the query set to dictionary clips corresponding to similar words and synonyms to words in the subtitles.

- P refers to new sign annotations obtained in this work through pseudo-labelling. In fact, we train a large-vocabulary (8K) sign classification model with automatic annotations from mouthings (M), dictionaries (D) and attention (A) and use it to pseudo-label. We firstly predict a sign class at each time step in a continuous signing video clip. We then filter the predicted signs to words in the corresponding subtitle.

- E and N are automatic sign annotations obtained in this work by relying on in-domain occurrences of signs. We localise a sign w in a reference video V_0 given (i) the word corresponding to w occurs in the subtitle associated with V_0, and (ii) other exemplar videos $V_1 \dots V_N$ with w in the associated subtitles. When mining instances of *known* classes E, the exemplar videos $V_1 \dots V_N$ are short video segments of the sign w from previous annotation methods. When mining instances of *novel* classes N, the exemplar videos $V_1 \dots V_N$ are longer, subtitle-length videos that have w in their corresponding subtitle. E and N are collected by calculating a matrix of cosine similarities between video features of the reference and exemplar videos. These video features are extracted from the last layer of a sign recognition MLP model trained with M*, D*, A [10], and P (see Tab. 4 in main paper). These in-domain methods are necessary as not all signs have mouthing cues, and signs in continuous signing may differ from their isolated realisations in dictionaries.

B Implementation Details

B.1 Mining more Spottings through In-domain Exemplars (E)

To mine for in-domain exemplars as described in Sec. 3.1, we choose N video exemplars of spottings of signs that we wish to find in the reference video. For an exemplar sign, we choose 8 consecutive stride-4 features surrounding each spotting ($|\mathcal{C}_i| = 8$ for $i = 1 \dots N$), where the features come from the last layer of the M* + D* + A [10] + P MLP model of Tab. 5 of the main paper and are 256 dimensional. The values of N are shown in the fourth column of Tab. 3. For the reference video, we choose a subtitle with 2s padding on either side, and use stride-4 features as candidate locations of signs.

The methods 'avg' and 'max' noted in the fifth column of Tab. 3. are computed slightly differently to the method 'vote' described in Sec. 3.1. As before, we compute the cosine similarity between each feature at each position of the reference video $c_0 \in \mathcal{C}_0$ and each position of the spottings exemplars $(c_1, \dots, c_n) \in \mathcal{C}_1 \times \dots \times \mathcal{C}_N$. The cosine similarity is rescaled to the interval $[0, 1]$. This results in N score maps of dimension $|\mathcal{C}_0| \times |\mathcal{C}_i|$ for $i = 1 \dots N$, which for us can be represented as a matrix $\mathcal{M}$ of dimension $|\mathcal{C}_0| \times 8 \times N$. We take either the average or the maximum value of $\mathcal{M}$ over the N exemplars to obtain a matrix $\mathcal{M}'$ of dimension $|\mathcal{C}_0| \times 8$. We then take the maximum of $|\mathcal{C}_0| \times 8$ across the exemplar temporal dimension to obtain a vector L of dimension $|\mathcal{C}_0|$. We consider the first element of L above a threshold h to be the corresponding sign in the reference video. For the version where we take the average value of $\mathcal{M}$ over the N exemplars, we let $h = 0.7$; for the version where we take the maximum value of $\mathcal{M}$ over the N exemplars, we let $h = 0.8$.

B.2 Discovering Novel Sign Classes (N)

In order to find a sign corresponding to a word w in a reference video, we take $N = 9$ positive exemplars corresponding to subtitles containing w, and $N' = 27$ negative exemplars corresponding to subtitles not containing w. We do not use padding around either the reference video nor the exemplars. The confidences for these spottings correspond to the proportion of the N exemplars with a cosine similarity match above a threshold h, i.e. the maximum value of L^+ as described in Sec. 3.2. We consider all novel sign classes with a confidence threshold above 0, that is, with at least one match amongst the positive exemplars.

B.3 Synonym Collection

We use synonyms both when querying keywords for spottings and when evaluating the performance of our MLP model. For these two purposes, we construct two different lists of synonyms. The first list is used for querying keywords for spottings and is large and flexible. The second list is a subset of the first; it is used to deem a prediction correct when evaluating our MLP model and is therefore more restrictive.

The first list is an extensive list of synonyms combined from multiple sources: the online dictionaries SignBSL[5], and BSL SignBank[6] 'related words' propositions for each sign video entry; words from the English synonym list from WordNet [4] as well as words with GloVe [7] cosine similarity above 0.9. In order to reduce noise, we remove synonyms with GloVe cosine similarity of less than 0.5. The second list of synonyms is a subset of the first, but we do not add all words with GloVe [7] cosine similarity above 0.9. Instead, amongst words with GloVe similarity above 0.9, we keep only those predicted to be sign synonyms by a simple sign synonym detection model. The sign synonym model is a 4 layer MLP model predicting whether or not two video features correspond to the same or different signs. The model is trained on pairs of M+D+A spottings from [1], and evaluated using the validation split with 33 videos, rather than the 36 aligned test set episodes used in the rest of the paper. At evaluation, we search for sign synonyms from our first list only amongst words with GloVe similarity of 0.9 and above. For each potential pair or synonyms with more than 5 spottings in the evaluation set, we consider the pair to be sign synonyms if it is predicted to be identical for at least 50% of the evaluation set examples. Tab. A.1 shows examples of synonyms.

Table A.1: **Examples of synonyms**: Our list of synonyms contains English words with similar meaning or words that can be signed using the same sign.

Word	**Synonyms**
change	evolution, diversity, conversion, switch, variety, convert, other, acquire, transform, amend, transformation, deepen, selection, evolve, adaptation, alteration, amendment, various, adapt, transfer, become, exchange, alter, modify, variation, modification, vary, among, shift
bus	coach, heap, metro, subway, tube, underground, vehicle, bus stop
rare	uncommon, few
content	message, capacity, substance, subject, context, insert, relief
architect	designer
airplane	aeroplane
skyscraper	city
king	royal, prince, princess, mogul, queen, power, tycoon, baron

B.4 Transpotter Finetuning

In Section 3.4 of the main paper, we discuss the domain gap between the lip movements in videos with the audio track removed (for example, from TV programmes) and the mouthings in sign language videos. As the Transpotter [8] is trained on the former, we finetune it on the pseudo-annotated sign language

[5] www.signbsl.com
[6] bslsignbank.ucl.ac.uk

mouthings to reduce the domain gap. In this section, we describe the process of extracting pseudo annotations and the subsequent finetuning.

Extracting Pseudo-annotations: We start with a pre-defined list of keywords that are at least 3 phonemes in length according to the CMU dictionary [9] and find all occurrences of these keywords in the subtitles. We take the video segment corresponding to the subtitle as our search window. We add 10 second padding (as also done in [1]) on either side of this video segment to account for the temporal misalignment between the continuous signing and audio-aligned subtitles. We query for the keywords present in the subtitle in order to obtain the temporal localization of each keyword in the video segment. As the video segment is much longer than the segments seen by the model during training, we perform a windowed inference with short 3 second windows. We have a 1.2 second overlap between successive windows. We run two windowed passes through the video, where the start time of the second pass is delayed by one second. This is to ensure that in at least one of these passes, the desired sign (often < 1 second in length) occurs completely within the short window. The Transpotter outputs a per-frame probability indicating whether a word is uttered at that frame. We save the frame number with the maximum probability as a possible annotation for the word and later filter these annotations based on confidence values.

Finetuning: As described in the main paper, we perform two rounds of finetuning. We first extract pseudo-labels using the Transpotter model from [8], pretrained on silent speech videos. We filter the mouthings with a confidence ≥ 0.7 as positive samples. In each batch, we oversample negative word-video pairs, in order to reduce false positives. We finetune the pre-trained Transpotter at a low learning rate of $1e^{-6}$ using the AdamW optimizer [5]. After convergence, we extract annotations with this more accurate finetuned model. We finetune the model a second time using the same hyper-parameters as above but resuming from the model weights from the first stage of finetuning. Further rounds of finetuning bring negligible improvements. Our final mouthing annotations M^* are extracted using this model.

How Does Finetuning Help? We observe that the Transpotter pre-trained on silent speech segments produces a large number of false detections on signing video segments as shown in Fig A.1.

After finetuning, the model is less likely to erroneously predict a query word. The decreased number of false positives is reflected by a reduction in overall size of the automatically annotated dataset, noted in Tab. 2 of the main paper. The finetuned model only spots $412K$ mouthings compared to the pre-trained model's $661K$. Despite a $1.5\times$ reduction in dataset size, the MLP model achieves better performance when trained on the $412K$ mouthings. Thus, finetuning the Transpotter improves downstream task performance, while also enabling faster and more efficient training of our MLP classifier due to fewer training samples.

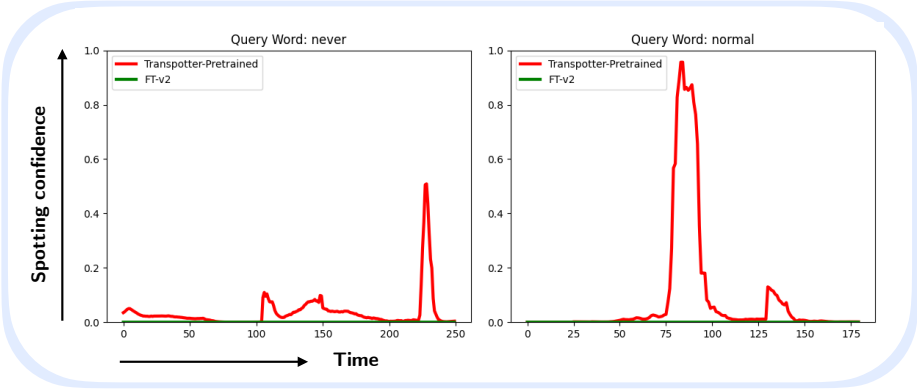

Fig. A.1: **Finetuning the Transpotter on pseudo-annotations leads to fewer false positive detections**: We show two qualitative examples to illustrate the impact of finetuning the mouthing model. For a given query word and a short video segment, we plot the per-frame confidence scores of the two models, i.e. before and after finetuning. We can see that the pre-trained Transpotter spots mouthings even though they are not present, whereas the finetuned model correctly predicts near-zero confidence, indicating that the word is indeed not mouthed in the given video segments.

B.5 Video Backbone (I3D)

Here, we describe the training of our I3D video backbone, which is used as the frozen feature extractor and as the source of pseudo-labelling for sign spotting.

As shown in Tab. A.2, we start with the M+D baseline from [1] which is initialised with Kinetics [3] pretraining. This model is trained with sign annotations with confidence above 0.8, resulting in 426K training samples from a 2,281 sized vocabulary. The model takes as input 16 consecutive video frames at 25 fps and a cropped 224×224 spatial region (from an initial 256×256 region). The input to the model is therefore $3 \times 16 \times 224 \times 224$, since our frames are RGB. For each sign annotation from mouthing (M), a sequence of 16 contiguous frames is randomly sampled from a window covering 15 frames before the time associated with the annotation and 4 frames after the annotation, i.e., $[-15, 4]$ around the mouthing peak. For dictionary annotations, the window around the similarity peak is $[-3, 22]$. I3D is trained for 25 epochs using SGD with momentum (with a momentum value of 0.9), with a batch-size of size 4. An initial learning rate of 0.01 is decayed by a factor of 10 after 20 epochs. Augmentations are applied during training including spatial cropping and color augmentations as well as scale and horizontal flip augmentations. The model produces a 1024-dimensional embedding (following average pooling) which is passed to our last linear layer, which outputs scores with the dimensionality of the number of classes. When evaluating the I3D predictions, I3D is run in a sliding window manner over the continuous signing with a stride of 4.

Table A.2: **Video backbone (I3D)**: We highlight the improved performance of I3D on the test set (SENT-TEST) when trained on a larger vocabulary (8K instead of 2K) and better pretraining using BSL1K [2, 10].

Annot. source	Pre-training	Num. I3D train annot.	Vocab. size	I3D predictions (subtitle independent)		
				Recall	IoU	Coverage
M [2]+D [6] [1]	Kinetics [3]	426K	2K	25.3	6.3	15.4
M [2]+D [6]	Kinetics [3]+BSL1K [2, 10]	426K	2K	25.5	6.4	15.5
M [2]+D [6]	Kinetics [3]+BSL1K [2, 10]	670K	8K	**26.3**	**7.9**	**16.3**

We explore how changing our pretraining effects performance: instead of only pretraining on Kinetics, we use a publicly released model (available on the webpage for [10]) which is first pretrained on Kinetics then finetuned on BSL1K [2] on a 5K vocabulary size. As shown in Tab. A.2, this marginally improves performance on our downstream task of continuous sign recognition.

We explore how expanding the vocabulary from 2K to 8K varies performance: this increases the number of training instances with confidence over 0.8 from 426K to 670K. In this case, our model is only trained for 17 epochs (due to computational costs) with an initial learning rate of 3e-2, reduced by a factor of 10 at epoch 12. As shown in Tab. A.2, this increases our recall from 25.5 to 26.3 and coverage from 15.5 to 16.3. This final model is chosen as our frozen feature extractor and as our source of pseudo-labelling for sign spotting: both features and class predictions are obtained by running I3D in a sliding window fashion with a stride of 4.

B.6 Lightweight Classifier (MLP)

As new sets of spottings are generated, a light weight MLP classifier is trained on the pre-extracted I3D features. Our 4-layer MLP module has layers of dimension (1024,512,256,8K) where the last layer corresponds to the number of sign classes and contains LeakyRelu activations in between. The first linear layer also has a residual connection on the 1024-dimensional I3D input features. The MLP is trained with a batch size of 128 for 15 epochs, with the learning rate initially set to 1e-2 and decayed by a factor of 10 at epochs 5 and 10. When evaluating the MLP predictions, the MLP is run in a sliding window fashion, outputting one feature for each I3D input feature (where the I3D features are extracted with a stride of 4).

C Additional Experiments

C.1 Varying Spotting Confidence

We show how varying the spotting confidence impacts the quality of spottings from previous methods, versus the improved spottings proposed in our work. As shown in Tab. A.3, even when reducing the confidence, our improved M^* + D^* give the best performance on our downstream task.

Table A.3: **Varying spotting confidence:** We highlight how varying the spotting confidence changes both our Spotting and MLP prediction evaluation performance. We evaluate on the test set (SENT-TEST).

	Training set			**Spottings [full] (subtitle dependent)**			**MLP predictions [8K] (subtitle independent)**		
Annotation source	full vocab	#ann. [full]	#ann. [8K]	Recall	IoU	Coverage	Recall	IoU	Coverage
M(0.8) + D(0.8)	15.0K	680K	670K	8.5	8.3	4.8	24.9	7.1	15.5
M(0.8) + D(0.75)	15.9K	1.90M	1.76M	18.4	16.9	9.8	23.3	8.1	15.1
M(0.8) + D(0.7)	16.6K	5.22M	4.86M	33.1	29.2	16.6	18.9	7.8	13.4
M(0.5) + D(0.7)	24.7K	5.74M	5.32M	35.3	30.9	17.5	19.1	7.8	13.4
M(0.5) + D(0.7) + A(0)	24.7K	6.17M	5.74M	37.0	32.5	18.3	20.3	8.3	13.9
M*(0.8) + D*(0.8)	20.9K	2.00M	1.94M	19.0	17.6	10.5	29.0	7.9	18.4
M*(0.8) + D*(0.75)	21.7K	7.89M	7.77M	41.9	36.7	24.0	27.0	7.7	18.5
M*(0.5) + D*(0.75)	22.4K	7.97M	7.84M	42.4	37.1	24.3	27.2	7.8	18.6
M*(0.5) + D*(0.75) + A(0)	22.5K	8.40M	8.28M	43.8	38.3	24.8	27.5	7.9	18.7

D Qualitative examples

D.1 Densification Visualisations

In Fig. A.2, we show visualisations of our densified sign sequences after our framework is applied.

D.2 Known Classes Spottings Visualisations

In Fig. A.3, we show visualisations of our score maps for annotating instances of known classes through our in-domain exemplar signs.

D.3 Novel Classes Spottings Visualisations

We show visualisations of score maps for annotating instances of novel classes through our in-domain weak exemplar subtitles. Fig. A.4 illustrates the necessity of using negative samples to avoid incorrectly identifying signs common to many subtitles such as pointing signs, pause gestures or other common gestures as the common lexical sign across exemplars. Fig. A.5 shows a failure case, where we cannot identify the sign for 'mandible' due to two different realisations of the sign depending on context.

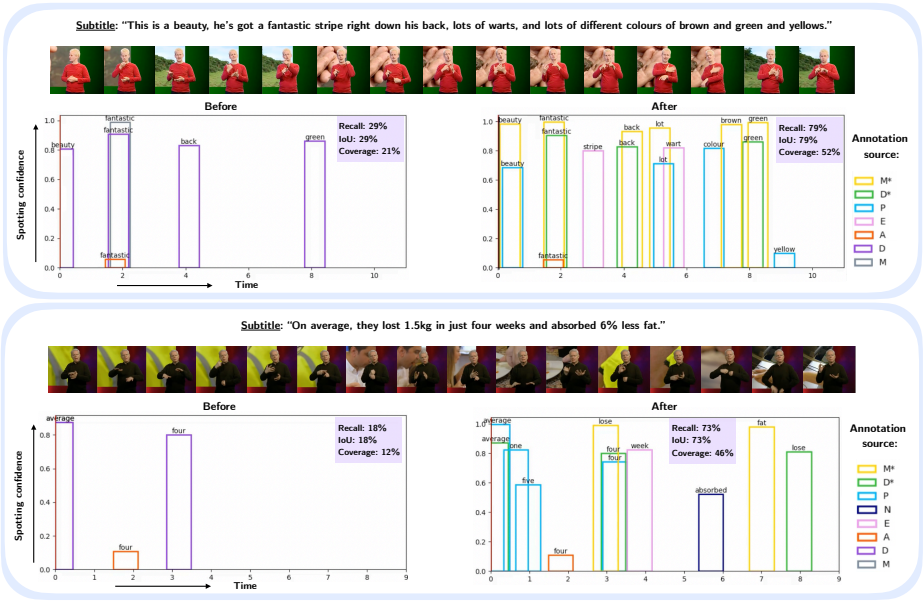

Fig. A.2: **Densification**: For two continuous signing sequences, we show plots of automatic sign annotation timelines, along with their confidence and annotation source, *before* and *after* our framework is applied. We observe that our method enables *densification* by two measures: removing gaps in the timeline so that we have a dense signing sequence spotted; and also increasing the number of words in the corresponding spoken language subtitle we recall. M, D, A refer to spottings obtained from previous methods from mouthings [2], dictionaries [6] and attentions [10] respectively. M^*, D^*, P, E, N refer to new and improved spottings from mouthings [8], dictionaries [6], I3D sign recognition pseudo-labels, in-domain exemplar spottings of known sign classes as well as in-domain exemplar spottings of novel classes respectively.

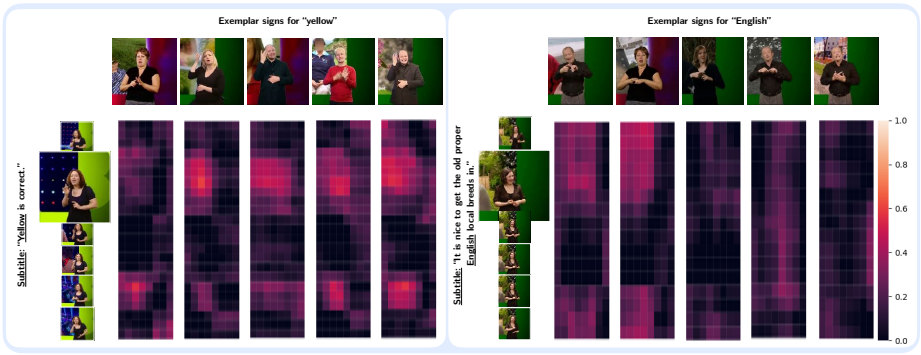

Fig. A.3: **Mining with spotting exemplars**: By comparing the score maps between a subtitle text and multiple spotting exemplars, we can temporally locate a lexical sign in a video segment. The left example illustrates how we can find the sign for 'yellow'. There are two different signs for 'yellow', where the second, third and fifth exemplars correspond to the sign used in the subtitle, and the first and fourth exemplars show an alternative sign. By using a voting method, we can count the number of exemplars with a high cosine similarity at a particular temporal location in the reference subtitle. The right example searches for the sign 'English' in a subtitle using 5 exemplars. The fifth exemplar in an incorrect spotting annotation, and has a low cosine similarity. However, with enough exemplars by different signers in different contexts, we can locate likely temporal locations of a sign in a subtitle.

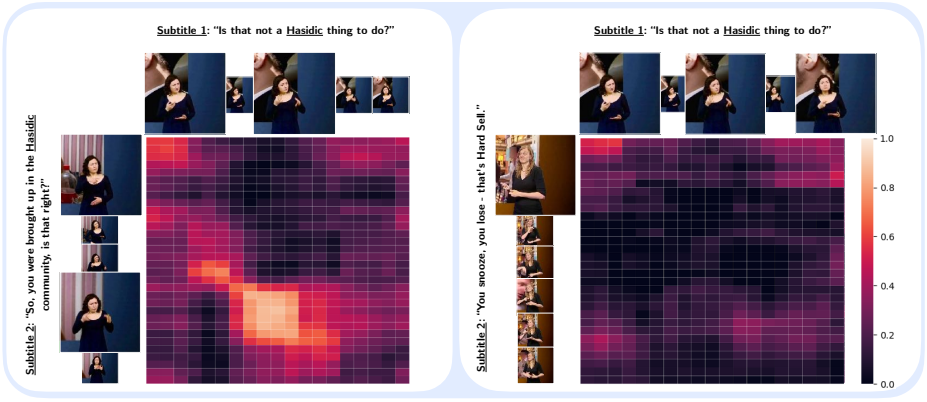

Fig. A.4: **Necessity of negative samples**: On the left, we show the cosine similarity between features of two subtitles, both of which contain the word 'Hasidic'. The cosine similarity is indeed high at the temporal intersection of both signs for 'Hasidic'; but the cosine similarity does also peak at pointing signs common to both subtitles. On the right, we show a score map for a subtitle containing the word 'Hasidic' and a subtitle without this keyword. By using the score maps of negative examples, we can identify non-lexical signs common across subtitles, such as pointing signs, and hence avoid incorrectly labeling the common lexical query sign.

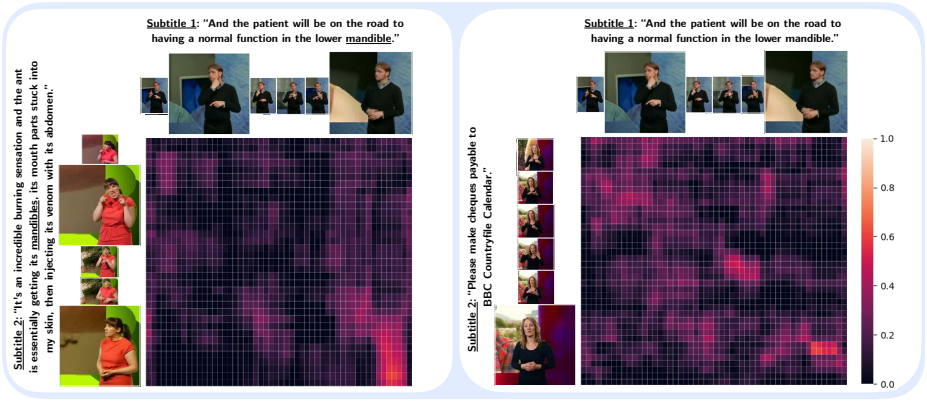

Fig. A.5: **Failure case**: On the left, we show the score map for two subtitles sharing the common word 'mandible'. However, in the first example, 'mandible' refers to a human mandible and in the second example, mandibles of an ant. The sign language interpretation of this word differs in each context, and the score map only shows strong cosine similarity when the signers are in a neutral pause position. The right score map demonstrates that this neutral position, frequent across many subtitles, can be located using negative exemplars. Using information from negative exemplars, we can avoid incorrect annotations.

Bibliography

[1] Albanie, S., Varol, G., Momeni, L., Afouras, T., Bull, H., Chowdhury, H., Fox, N., Cooper, R., McParland, A., Woll, B., Zisserman, A.: BOBSL: BBC-Oxford British Sign Language Dataset. arXiv preprint arXiv:2111.03635 (2021) 4, 5, 6, 7

[2] Albanie, S., Varol, G., Momeni, L., Afouras, T., Chung, J.S., Fox, N., Zisserman, A.: BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In: ECCV (2020) 1, 7, 9

[3] Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 6, 7

[4] Feinerer, I., Hornik, K.: wordnet: WordNet Interface (2020), https://CRAN.R-project.org/package=wordnet, r package version 0.1-15 4

[5] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 5

[6] Momeni, L., Varol, G., Albanie, S., Afouras, T., Zisserman, A.: Watch, read and lookup: Learning to spot signs from multiple supervisors. In: ACCV (2020) 1, 7, 9

[7] Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014) 4

[8] Prajwal, K., Momeni, L., Afouras, T., Zisserman, A.: Visual keyword spotting with attention. In: BMVC (2021) 2, 4, 5, 9

[9] Speech Group at Carnegie Mellon University: CMU pronouncing dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict (2014) 5

[10] Varol, G., Momeni, L., Albanie, S., Afouras, T., Zisserman, A.: Read and attend: Temporal localisation in sign language videos. In: CVPR (2021) 2, 3, 7, 9