Proceedings of the 9th Workshop on the Representation and Processing of Sign Languages, pages 217–223
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Cross-Lingual Keyword Search for Sign Language

Nazif Can Tamer, Murat Saraçlar
Boğaziçi University
Department of Electrical and Electronics Engineering
{can.tamer, murat.saraclar}@boun.edu.tr

Abstract

Sign language research most often relies on exhaustively annotated and segmented data, which is scarce even for the most studied sign languages. However, parallel corpora consisting of sign language interpreting are rarely explored. By utilizing such data for the task of keyword search, this work aims to enable information retrieval from sign language with the queries from the translated written language. With the written language translations as labels, we train a weakly supervised keyword search model for sign language and further improve the retrieval performance with two context modeling strategies. In our experiments, we compare the gloss retrieval and cross language retrieval performance on RWTH-PHOENIX-Weather 2014T dataset.

Keywords: sign language recognition, cross language retrieval, context modeling, weakly supervised learning, attention

1. Introduction

Most of the existing data in sign language comes from the public media, where one finds news, shows, TV series, and movies interpreted for the Deaf in sign language. Although the amount of data available in this format is great in scale, these parallel corpora are often considered too noisy and unreliable for sign language research. The grammar and word ordering of the written/spoken language and the corresponding sign language interpreting do not match one to one, and thus, translations in the written language cannot be directly used to train current automatic recognition systems that require at least ordered glosses as the label.

Since the effective utilization of these parallel corpora would greatly increase the overall number of data available for sign language studies, researchers actively try to convert this weakly supervised and noisy data into a more convenient format for research. Pfister et al. (2013) and Kelly et al. (2011) use the weak supervision coming from the translations to automatically extract isolated signs with multiple instance learning (MIL) based strategies, and train models with the segmented data. In a different direction, Camgoz et al. (2018) dropped segmentation out of the equation and applied state-of-the-art neural machine translation approaches to directly translate sign language videos to the written language in an end-to-end manner. In this work, by utilizing a similar strategy, we train an end-to-end keyword search model by searching the sign language sentence for words coming from the translations in written language.

Although keyword search is a new application for sign languages, it is a well-studied problem for spoken languages. The most common strategy is to use lattices generated by automatic speech recognition (Saraclar and Sproat, 2004). More recently, end-to-end keyword search strategies also started to appear (Audhkhasi et al., 2017). We previously used end-to-end methods for gloss search from sign language videos in (Tamer and Saraçlar, 2020), and this work is an extension of that.

The main contribution of this work on top of our previous model is the introduction of the context modeling for cross-lingual keyword search. Rescoring keyword search predictions with the predictions for other keywords (Karakos et al., 2013) and the predictions of the same keyword at another close time instant (Richards et al., 2014) is a known strategy in spoken keyword search. In this work, we apply this rescoring strategy to our model's cross-lingual keyword search and show that model's own predictions for other keywords can be used to boost keyword search performance.

The rest of this paper is organized as follows. In Section 2, the previously-introduced end-to-end keyword search network is summarized briefly. In Section 3, the modifications made specifically for cross-lingual search is explained. In Section 4, the dataset and evaluation metrics are given. Lastly, in Section 5, in addition to giving our results for keyword search and comparing them to gloss search, we further discuss how this weakly supervised training strategy helps automatic segmentation of parallel corpora between written language and sign language interpreting.

2. Weakly Supervised Keyword Search for Sign Language

The model structure is summarized in Figure 1. After the video is converted into a sequence of skeleton joints, the rest of the keyword search model is trained end-to-end by searching for text or gloss queries in the sign language sentence. In short, the aim of this training strategy is to represent both a query and the relevant part of the sign language sentence by a similar vector in a mutual latent space. This is done by the joint training of spatio-temporal graph convolutional network (ST-GCN) encoder, word embedding, and the attention based selection mechanism.

2.1. ST-GCN Encoding of the Sign Language Sentence

Spatial Temporal Graph Convolutional Networks (Yan et al., 2018) first introduced for the skeleton-based action recognition is used for the encoding of the skeleton sequence. In this model, a graph connecting neighboring skeleton joints and the same joints across frames (see the

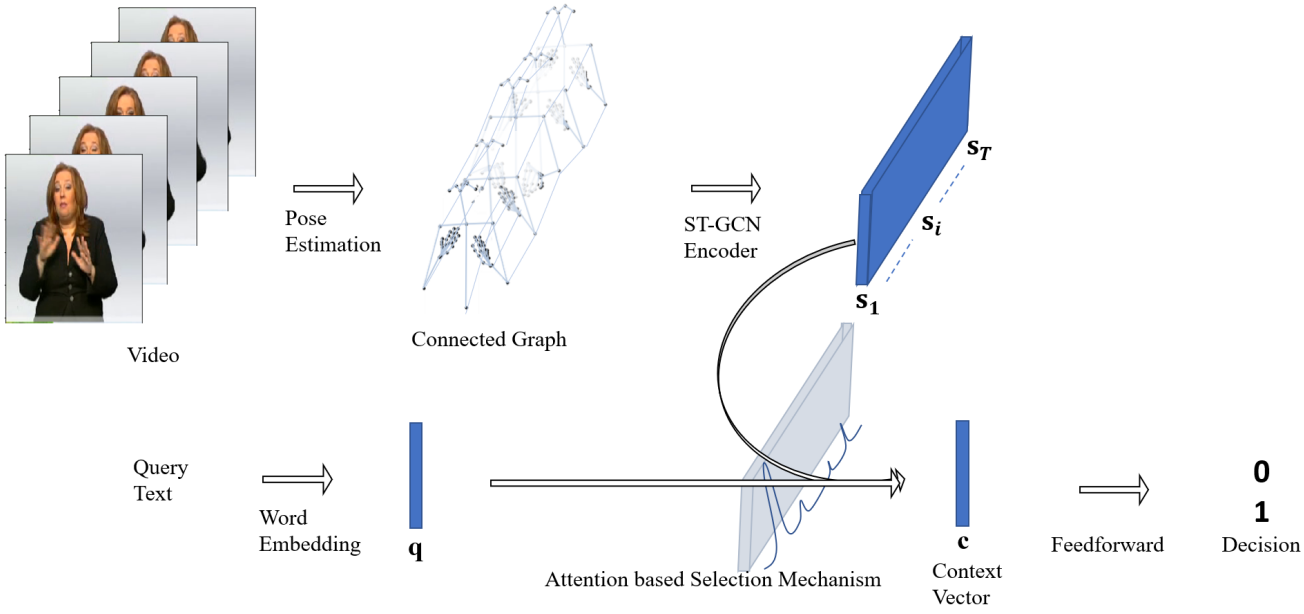

Figure 1: After pose estimation with OpenPose, the rest of the framework is trained end-to-end.

connected graph in Figure 1) is formed and 12 layers of graph convolution operations take place on top of this connected graph as described in our previous work (Tamer and Saraçlar, 2020).

2.2. Query Embedding and the Attention-based Selection Mechanism

Query embeddings, vectors representing each query in our vocabulary, are learned through attention based selection mechanism. Let $\mathbf{q}$ represent the word embedding for the query and $\mathbf{s}_i$ the ith member of the encoded sign language sentence; the similarity score between $\mathbf{q}$ and $\mathbf{s}_i$ is obtained for all $i \in (1, T)$ through the scoring function

$$score(\mathbf{q}, \mathbf{s}_i) = \beta \left[\frac{\mathbf{q} \cdot \mathbf{s}_i}{\|\mathbf{q}\| \cdot \|\mathbf{s}_i\|} \right]^2 + \theta \qquad (1)$$

with β and θ learnable parameters. From the similarity scores, a single context vector $\mathbf{c}$ is obtained

$$\mathbf{c} = \sum_i \left[\frac{\exp\left(score(\mathbf{q}, \mathbf{s}_i)\right)}{\sum_{i'} \exp(score(\mathbf{q}, \mathbf{s}_{i'}))} \right] \cdot \mathbf{s}_i \qquad (2)$$

and a simple fully connected layer decides on whether the query $\mathbf{q}$ is found in the entire $\mathbf{s}_{1:T}$ sequence.

2.3. Training Strategy

The stability of the training is ensured by searching for all the queries in our vocabulary in the same sign language sentence. With the skeleton sequence and all the queries in our vocabulary at the input, the network is trained to minimize the binary cross entropy loss between its predictions for each query and the labels obtained by simply giving 1 if the query is in the translation and 0 if it is not.

3. Query-Specific Context Modeling for Cross-Lingual Keyword Search

The motivation behind query-specific context modeling is that, when doing cross-lingual retrieval, words in the spoken language do not match one to one with their sign language glosses. Furthermore, for some less frequent words, the predictions we get by training with only small amount of data are most often unreliable. To remedy these problems, we define two prediction rescoring strategies that use model's own predictions for other queries within the vocabulary V. For a single sign language sentence, let $\vec{l}$ represent the $|V|$ dimensional correct labels, s.t. we have one label $\in \{0, 1\}$ for each query, and $\vec{p}$ represent the $|V|$ dimensional vector comprised of the trained model's predictions. Our aim is to come up with a new predictions vector $\vec{p}$ that is better than the original $\vec{p}$. We do this by two different strategies: (i) a statistical context model based on bag-of-words TF-IDF vectors, and (ii) a machine-learning based multi-layer perceptron (MLP) context model.

3.1. Statistical Context Modeling with TF-IDF Vectorization

Term Frequency Inverse Document Frequency (Ramos and others, 2003) vectorization is a well known strategy for language modeling. While calculating a weight for a query inside a document, this algorithm gives high weights to queries that are seen multiple times in this document (high term frequency), and low weights to ones that are seen in many other documents (inverse document frequency). Thus, by finding the similarity between each keyword in the vocabulary V and the document, we obtain a $|V|$ dimensional vectoral representation of the document. Let $\vec{d}_i$ be the l_1 normalized TF-IDF vector for the ith document in our training set, the document context model $\vec{d}_q$ for query q is found by averaging over all the documents in our training set that contain this specific query:

$$\vec{d}_q = \text{avg}(\vec{d}_i : \text{tfidf}(q, \vec{d}_i) > 0) \qquad (3)$$

Then, by looking at the cosine similarity between this query-specific document context vectors $\vec{d}_q$ and our

model's prediction vector $\vec{p}$, we obtain the new scores. The new prediction score for the ith query in the vocabulary $\tilde{p}(q_i)$ is formulated as

$$\tilde{p}(q_i) = 1 - \frac{\vec{d_q} \cdot \vec{p}}{|\vec{d_q}||\vec{p}|} \tag{4}$$

and the new prediction vector $\vec{\tilde{p}}$ is simply the new prediction values for all the queries in our vocabulary V.

$$\vec{\tilde{p}} = \left[\tilde{p}(q_1), ..., \tilde{p}(q_i), ..., \tilde{p}(q_{|V|})\right]^\intercal \tag{5}$$

3.1.1. Fusion Strategy

The statistical context modeling for the query, by itself, does not give better results than the model's own predictions. However, when combined with the original predictions through a hyperparameter, it boosts the prediction scores. With $\vec{p}$ being the model's original predictions and $\vec{\tilde{p}}$ the predictions obtained through query context modeling, the final predictions are obtained by combining the two predictions using a hyper-parameter γ:

$$\log \vec{\tilde{p}}' = \gamma \cdot \log \vec{p} + (1 - \gamma) \cdot \log \vec{\tilde{p}} \tag{6}$$

In our experiments, we tuned the hyperparameter γ to maximize the mean average precision (mAP) score in the development set. For different graph layout options, results given at Table 2 are with the γ values of 0.40 for the upper body only layout, 0.54 for the upper body and dominant hand combined, and 0.58 for all upper body, the dominant hand and the passive hand combined.

3.2. Multilayer Perceptron Based Context Model

For a sign language sentence in our training set, we trained a simple multi-layer perceptron with $|V|$ dimensional predictions vector $\vec{p}$ as the inputs and labels vector $\vec{l}$ as the target. The network is comprised of two hidden layers of size 256 with ReLU activations and a dropout probabilty of 20%. We finished the training with early stopping when the loss in the development set was not reducing any further.

4. Experimental Setup

4.1. Dataset

We used RWTH-PHOENIX-Weather-2014T dataset (Camgoz et al., 2018) to conduct our experiments. Recorded in 25 fps videos, the dataset includes weather forecasts in sign language, their sentence-level gloss transcriptions (without temporal alignments), and the translations into the German language. The main reason we used this dataset for our experiments is that, by including both gloss transcriptions in German sign language and corresponding translations in German, it offers a natural medium for comparing cross-lingual keyword search with gloss search.

The dataset is partitioned into 9.2 hours of training, 37 minutes of development and 43 minutes of test data. In order to use this dataset in keyword search task, we segmented the transcriptions and translations into constituent words and used them as our queries. In the gloss search, the vocabulary consists of 1085 glosses that are seen at least once in the training set and 398 of these are also seen at least once in the test dataset. Thus, we report our results from this shared vocabulary of **398** queries. Similarly, for the cross-lingual search, we have 2887 words in the training set and 942 of these are also shared in the test set and we report our cross-lingual results on this shared vocabulary of **942** queries.

To clarify the training procedure with an example, let us consider the sequence in Figure 6: When training a gloss search model, a 40-frame long sign language sentence is labeled with $-1-$ for 3 glosses: "nordost", "bleiben" and "trocken", and $-0-$ for the remaining 1082 glosses. When training a cross-lingual keyword search model, the same sequence is labeled with $-1-$ for 6 words: "im", "nordosten", "bleibt", "es", "meist", "trocken", and $-0-$ for the remaining 2881 words. A cross-lingual kws model cannot see the glosses and vice versa; gloss and cross-lingual search models are completely independent.

4.1.1. Skeleton Extraction from Video Frames

2D pose estimates of upper body, right and left hand are extracted through part affinity fields based OpenPose framework (Cao et al., 2017). In figure 2, you can see an example subsequence from a sign language sentence with OpenPose pose estimates projected on top. Since the frames are blurry and low resolution, the pose estimation process cannot always result in good (x, y) coordinate estimates for each joint. To remedy this, we also used the related confidence scores as the third dimension to feed into the graph convolutional encoder.

4.2. Evaluation metrics

For a query q, precision recall values at an operating point are defined as

$$\text{Precision} = \frac{|\{\text{Retrieved}\} \cap \{\text{Relevant}\}|}{|\{\text{Retrieved}|\}}$$

$$\text{Recall} = \frac{|\{\text{Retrieved}\} \cap \{\text{Relevant}\}|}{|\{\text{Relevant}\}|}$$

and precision-recall curve obtained at different operating points (e.g. by changing the threshold) is one of the most valuable metrics in evaluating the performance of information retrieval systems.

4.2.1. Term-averaged Precision-Recall Curve and the F1 Score

When precision and recall values associated with a threshold θ is averaged over different queries q, term-averaged precision-recall values are obtained for that threshold:

$$\text{Precision}(\theta) = \frac{1}{|Q|} \sum_{q \in Q} \text{Precision}(q, \theta)$$

$$\text{Recall}(\theta) = \frac{1}{|Q|} \sum_{q \in Q} \text{Recall}(q, \theta)$$

Thus, by sweeping through different θ thresholds, we obtain the term-averaged precision-recall curve that summarize the performance of the keyword search system. We also report the maximum of F1 scores summarizing the curve:

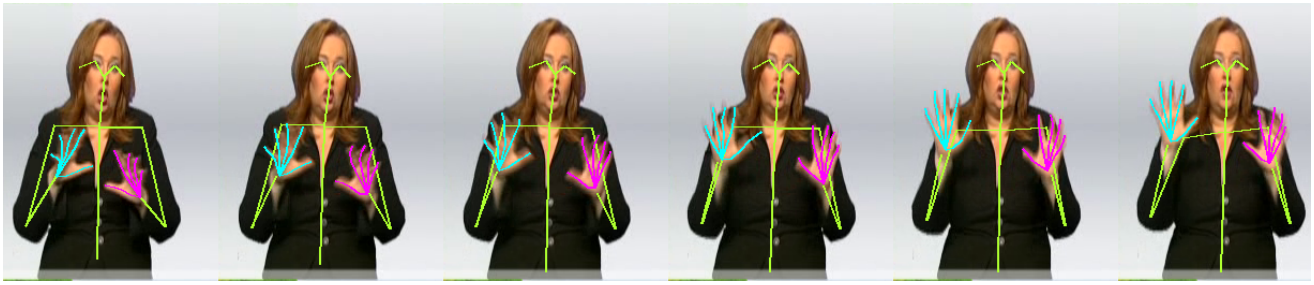

Figure 2: An example subsequence from the dataset. The extracted poses for upper body, the dominant hand, and the passive hand are shown on top of images in yellow, cyan, and magenta respectively. The poses constructed by OpenPose framework are highly representative even though the original input images are low resolution and blurry.

$$\text{maxF1} = \max_{\theta} \frac{2 \cdot \text{Precision}(\theta) \cdot \text{Recall}(\theta)}{\text{Precision}(\theta) + \text{Recall}(\theta)}$$

4.2.2. Mean Average Precision (mAP)

Similarly, in object and action recognition, one of the most used metrics is mean average precision. It roughly corresponds to the area under precision-recall curves belonging to different queries q averaged over queries.

$$\begin{aligned}\text{mAP} &= \frac{1}{|Q|} \sum_{q \in Q} \text{AveragePrecision}(q) \\ &= \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{|N|} \sum_{n=1}^{|N|} \text{Precision@}n(q)\end{aligned}$$

($|N|$: the number of relevant documents for query q). It is common to report mAP scores at different Intersection over Union (IoU) thresholds. However, since we did not have any labels for temporal alignments and segmentation, we simply report mAP scores with IoU=0.

5. Results and Discussion

In this section, we present our gloss and cross-lingual keyword search results obtained with different encoder graph structures and different context modeling strategies. We also compare cross-lingual KWS results obtained with a translation approach and visualize the temporal localization capabilities of our model.

5.1. Effects of Graph Layout: Upper Body, the Dominant and the Passive Hand

The upper body, the dominant hand, and the passive hand poses are all important components in understanding sign language. To identify the effects of different components in the performance of keyword search for sign language, we trained 3 gloss search and 3 cross-lingual keyword search models with the features in Figure 3.

From the results summarized in Table 1 and the precision-recall curve in Figure 4, we see that the upper body alone contains much of the information by itself. Introducing the dominant hand also significantly improves the results for both gloss and cross-lingual search models. However, we see that there is not much gain with the introduction of passive hand. Although the layout including the passive hand performs the best in cross-lingual search, it reduces both the mAP and maxF1 scores in the gloss search compared to the dominant hand + upper body layout option.

Since the OpenPose hand model has 21 joints, including the passive hand in the graph layout increases the number of graph nodes from 34 to 55 and demands more computational resources for graph convolution operations. Thus, we conclude that the costs of including the passive hand in the graph layout may outweigh the benefits.

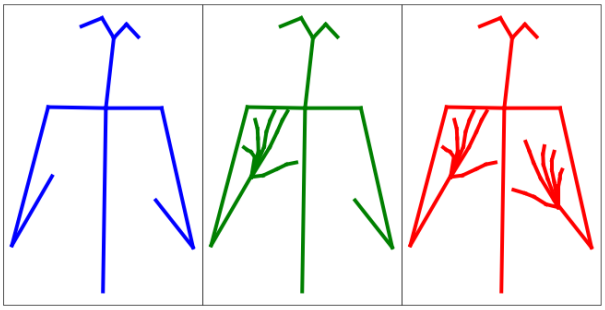

Figure 3: From left to right, the three graph layout options used in the experiments are upper body (13 joints), upper body with the dominant hand (34 joints), and upper body with both hands (55 joints), respectively.

	Gloss		Cross-Lingual	
	mAP (%)	maxF1	mAP(%)	maxF1
Upper Body (UB)	24.29	26.40	12.49	15.18
UB + Dom. Hand	**29.91**	**33.53**	13.18	15.62
UB + Both Hands	29.22	32.80	**14.56**	**16.15**

Table 1: Gloss and cross-lingual KWS results using two metrics (the higher the better, best scores for each task are in bold). Cross-lingual results are reported after MLP context model applied.

5.2. Effect of Different Context Modeling Strategies

Results obtained with different context modeling strategies are summarized in Table 2. Firstly, we can say that sta-

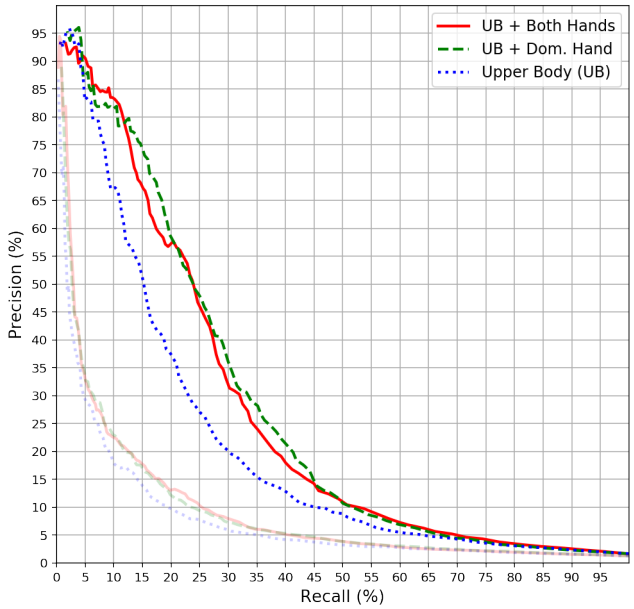

Figure 4: Precision-recall curves for the gloss search models with different layout options. The cross-language search results are shown in transparent for comparison.

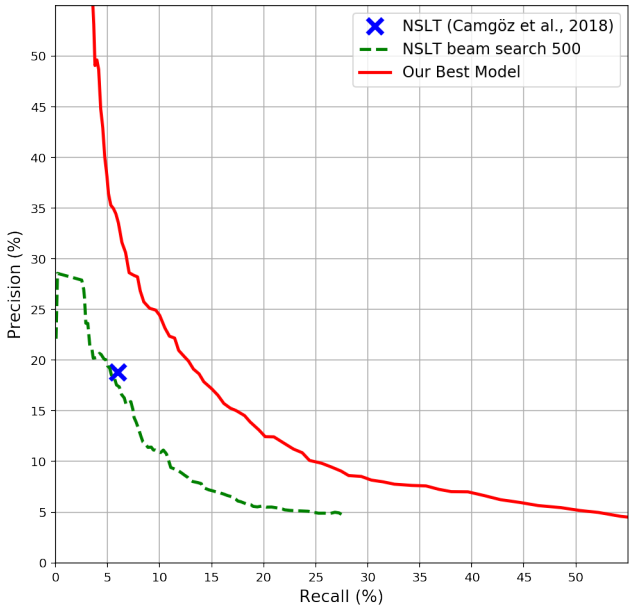

Figure 5: Our best cross-lingual KWS model (trained with UB + Both Hands layout option and MLP context model) compared to searching from Neural Machine Translation outputs (the higher the better).

tistical context modeling improves both metrics for all the layout options, and the gains are significant for the Upper Body + Both Hands layout. Secondly, we see that the MLP based context modeling did not improve the results for UB + Dominant Hand layout. Since we stopped the training of the MLP based context model when the development loss is not reducing any further, the results in the test dataset are not necessarily better. However,we obtained our best overall mAP score with an MLP based context model (with a significant increase from 13.01% to 14.56%).

5.3. Comparison to KWS from Neural Machine Translation Outputs

In spoken keyword search, a well-known strategy is to use the transcriptions obtained through automatic speech recognition (ASR). In a similar approach, we used translations obtained from Neural Sign Language Translation (Camgoz et al., 2018) model as our baseline. From the translations we get by using the same hyper-parameters in their paper, we obtain the single operation point denoted as NSLT in Figure 5.

In spoken keyword search, another strategy is to search for the keyword in lattices generated from ASR outputs (Saraclar and Sproat, 2004). Similarly, we plot precision-recall curve related to this NSLT model by applying beam search with beam size of 500 and finding the expected counts for each word along the beams. With the two as our baselines in Figure 5, we conclude that our cross-lingual KWS model is better than searching for keywords in translation outputs.

5.4. Temporal Localization as a By-Product of Weakly Supervised Training

When we have sequence-level, ordered gloss transcriptions of sign language data, HMM-based models can iteratively

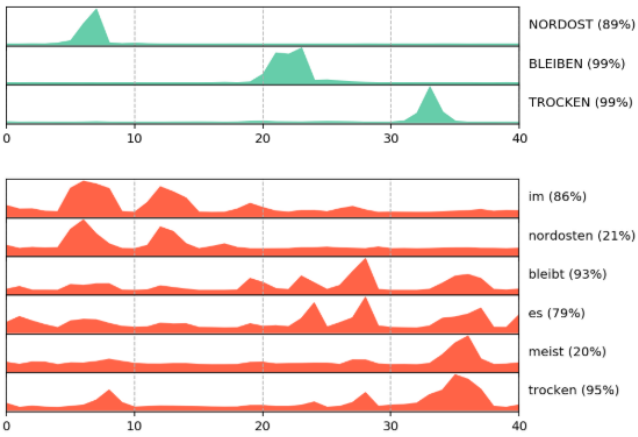

Figure 6: Temporal localizations for the sequence with gloss annotation “nordost bleiben trocken” and translation “im nordosten bleibt es meist trocken”. The prediction confidences are denoted in parentheses.

align each frame to a gloss hidden state and thus do the temporal segmentation as exemplified in (Koller et al., 2017). However, since these HMM models rely on the strictness of the order of a gloss sequence, this alignment procedure cannot work with the noisy and weak supervision of translations. In this section, we show that our model’s attention based selection mechanism can loosely localize some keywords independent of label type. For sign language sentences of varying length, we show the temporal keyword localization capabilities of our models that are trained with either gloss-sequences or translations as the labels.

In Figures 6, 7, and 8, we see model predictions (shown

	Without Context Model		Statistical C.M.		MLP-based C. M.	
	mAP (%)	maxF1	mAP (%)	maxF1	mAP (%)	maxF1
Upper Body (UB)	11.62	14.66	11.94	14.90	**12.49**	**15.18**
UB + Dom. Hand	13.66	16.10	**13.78**	**16.28**	13.18	15.62
UB + Both Hands	13.01	16.40	13.80	**16.69**	**14.56**	16.15

Table 2: Effect of context model on cross-lingual KWS. Best mAP and maxF1 scores for each layout are in bold, and overall best scores are underlined.

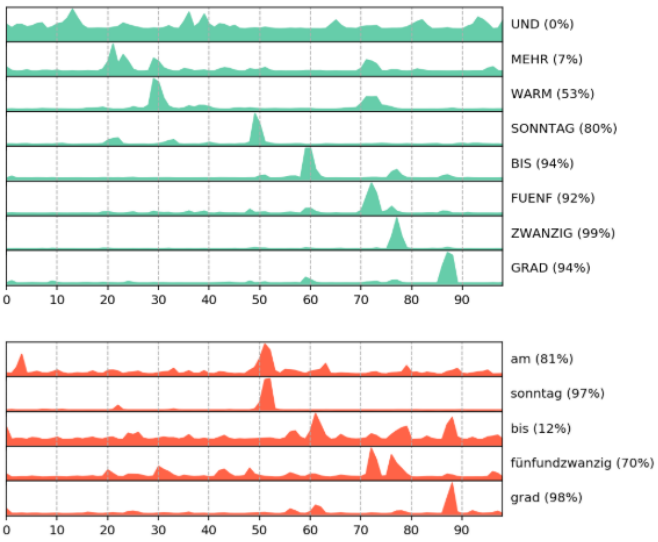

Figure 7: Temporal localizations for the sequence with gloss annotation "und mehr warm sonntag bis fuenf zwanzig grad" and translation "am sonntag bis fünfundzwanzig grad". The prediction confidences are denoted in parentheses.

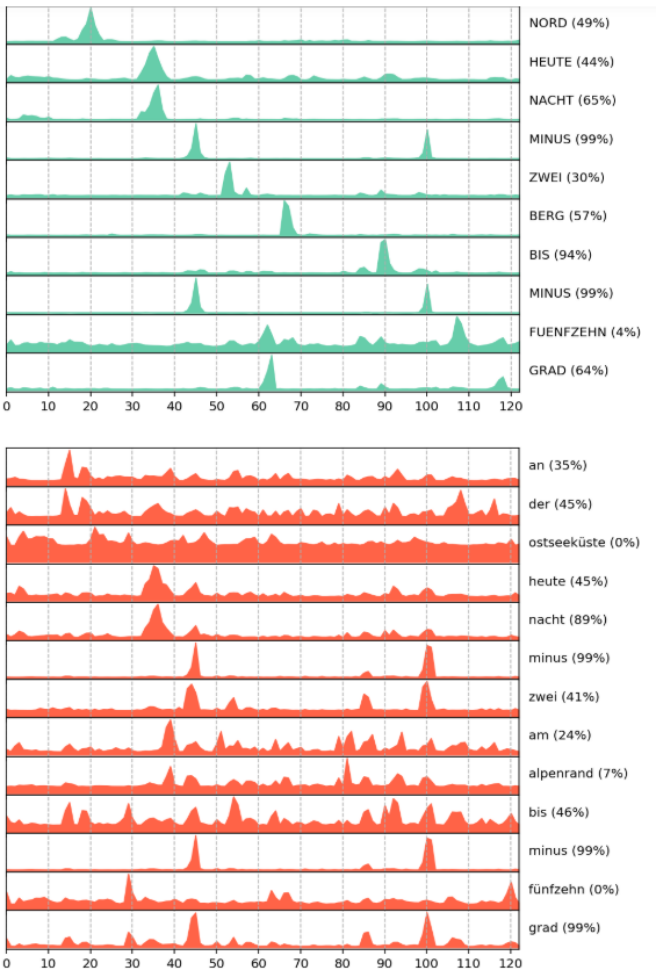

Figure 8: Temporal localizations for the sequence with gloss annotation "nord heute nacht minus zwei berg bis minus fuenfzehn grad" and translation ""an der ostseeküste heute nacht minus zwei am alpenrand bis minus fünfzehn grad". The prediction confidences are in parentheses.

with percentages next to the labels) and related temporal localizations (denoted by the most peaky regions) for both gloss and cross-lingual search. For the most of our data, we see that gloss search models are better in localization capacity and the order of peaky regions usually follows the gloss order correctly. We also see that peaky regions are more visible when the prediction confidences are higher. For the the cross-lingual search, we see that localization is possible for some words that are matching one-to-one with gloss transcriptions (such as "grad" in Figures 7 and 8, "sonntag" and "fünfundzwanzig" (with two peaks at both "fuenf" and "zwanzig") in Figure 7, "nacht" in Figure 8 etc.), but not so much for the conjugated verbs like "bleibt" in Figure 6, or words without a unique gloss such as "alpenrand" and "ostseeküste" in Figure 8. We believe that cross-lingual KWS is at least beneficial for finding the most salient temporal regions that might be related to any gloss.

6. Conclusion

In this paper, we employed a weakly-supervised, end-to-end training strategy for cross-lingual keyword search for sign language and showed that cross-lingual training is a viable option when we do not have the gloss labels. We introduced two context modeling strategies and further improved the cross-lingual keyword prediction results. We compared the retrieval performance and temporal localization capabilities of gloss and cross-lingual search under three different layout options. The most important contribution of this paper is the introduction of a cross-lingual KWS method that can theoretically utilize the widely available sign language interpretations in public media. In the future, we aim to apply the same strategy to bigger datasets.

7. Acknowledgements

This study was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under Project 117E059.

8. Bibliographical References

Audhkhasi, K., Rosenberg, A., Sethy, A., Ramabhadran, B., and Kingsbury, B. (2017). End-to-end asr-free keyword search from speech. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1351–1359.

Camgoz, C., Hadfield, S., Koller, O., Ney, H., and Bowden, R. (2018). Neural sign language translation. In *Proc. CVPR*, pages 7784–7793.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.

Karakos, D., Schwartz, R., Tsakalidis, S., Zhang, L., Ranjan, S., Ng, T., Hsiao, R., Saikumar, G., Bulyko, I., Nguyen, L., Makhoul, J., Grézl, F., Hannemann, M., Karafiát, M., Szoke, I., Veselý, K., Lamel, L., and Le, V. B. (2013). Score normalization and system combination for improved keyword spotting. pages 210–215, 12.

Kelly, D., McDonald, J., and Markham, C. (2011). Weakly supervised training of a sign language recognition system using multiple instance learning density matrices. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(2):526 –541, april.

Koller, O., Zargaran, S., and Ney, H. (2017). Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July.

Pfister, T., Charles, J., and Zisserman, A. (2013). Large-scale learning of sign language by watching tv (using co-occurrences). In *BMVC*.

Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ.

Richards, J., Ma, M., and Rosenberg, A. (2014). Using word burst analysis to rescore keyword search candidates on low-resource languages. pages 7824–7828, 05.

Saraclar, M. and Sproat, R. (2004). Lattice-based search for spoken utterance retrieval. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 129–136.

Tamer, N. C. and Saraçlar, M. (2020). Keyword search for sign language. In *Proc. ICASSP*.

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*.