Unsupervised Discovery of Sign Terms by K-Nearest Neighbours Approach

Korhan Polat[0000−0002−4397−0299] and Murat Saraçlar[0000−0002−7435−8510]

Boğaziçi University, Istanbul, Turkey
{korhan.polat, murat.saraclar}@boun.edu.tr

Abstract. In order to utilize the large amount of unlabeled sign language resources, unsupervised learning methods are needed. Motivated by the successful results of unsupervised term discovery (UTD) in spoken languages, here we explore how to apply similar methods for sign terms discovery. Our goal is to find the repeating terms from continuous sign videos without any supervision. Using visual features extracted from RGB videos, we show that a k-nearest neighbours based discovery algorithm designed for speech can also discover sign terms. We also run experiments using a baseline UTD algorithm and comment on their differences.

Keywords: unsupervised learning, sign language, term discovery

1 Introduction

Most of the automatic sign language recognition (ASLR) systems to date require large amounts of training data. Since there does not exist a reliable automatic annotation tool, sign corpora need to be annotated by human experts. Manual annotation being a laborious process, limits the number of available annotated corpora and hinders the development of better ASLR systems. However, there are plenty of sign language resources that can be used if we employ unsupervised learning methods. In this work, we address this issue and investigate how unsupervised term discovery in speech can be adapted to sign languages.

The aim of unsupervised term discovery (UTD) is to discover repeating units in an unknown language, without using any information except the signal itself (zero-resource). In general, UTD systems take feature time series as input and the output is the discovered clusters of segments, where each cluster is hypothesized to be a unit in that language. For spoken languages, the repeating units may correspond to phones, words or common phrases in that language. Usually UTD systems employ three stages. The first one is the matching stage, in which pairs of similar segments are discovered. The second stage involves the clustering of these pairs, so that similar pairs are joined together to form clusters of hypothesized units. The last stage concerns the parsing of the input sequences with discovered word-type IDs. The performance of the clustering and parsing stages depends on the quality of the matching stage. Hence, we narrow our scope to the matching stage only; our aim is to discover pairs of similar sub-sequences from continuous sign language videos, without any additional information.

Unsupervised term discovery has been studied in speech processing for over a decade. The pioneering work [18] in spoken term discovery introduces the segmental variant of dynamic time warping (SDTW) algorithm to search for pairs of similar segments. The input files are processed in pairs and pairwise distance matrices between their time series feature vectors are computed. The idea is to apply DTW in diagonal bands on a distance matrix between two sequences and collect the path fragments with minimal distortions. The discovered diagonal path fragments with high similarities are referred as the matching pairs, which are clustered to form hypothesized word categories. A similar but more efficient algorithm [4] uses locality sensitive hashing to approximate distance matrices. The diagonal fragments with high similarities are searched using efficient image processing techniques. Costly SDTW search is applied only in the vicinity of these candidate fragments, thus reducing runtime significantly. In the following years, Zero Resource speech challenges [25, 3] were held to allow comparison of various zero-resource approaches using standardized metrics. The Bayesian methods [17, 8] that perform full-coverage, require large amounts of data to be trained and assume that tokens of the same types do not show significant variability. We opt to use a simpler discovery method that requires no training, the K-nearest neighbours based algorithm [23]. We show that it can be run for continuous sign videos, by feeding visual features instead of speech features.

Unsupervised learning has been a rather inactive area in sign language recognition. Previous works that focus on lexicon discovery usually rely on weak supervision, in the form of subtitles [19] or text translations [9] that accompany sign videos. Other works that focus on extracting sub-units [26, 22, 24] do not perform discovery at sign level. A similar work to ours [16] finds common signs among continuous sentences, but uses the information that there is a common sign. These works rely on weak supervision or incorporate linguistic information to the discovery process. Zero resource term discovery for sign language is first explored in [20], in which the SDTW baseline algorithm [4] for speech is used to discover sign terms. Here, we build upon the same idea and show that a KNN based term discovery algorithm [23] can also be adapted for sign language term discovery. We also employ a better cross-validated evaluation scheme and compare our results to SDTW baseline in [20].

In short, our contribution is to show that a KNN based term discovery algorithm [23] can be used for sign languages. We also make a comparison with the SDTW based baseline in [20], while improving the evaluation scheme. In Section 2, an overview of the discovery pipeline [23] is given for the sake of completeness. In Section 3 the setup for sign language experiments are presented. Results are given and discussed in Section 4.

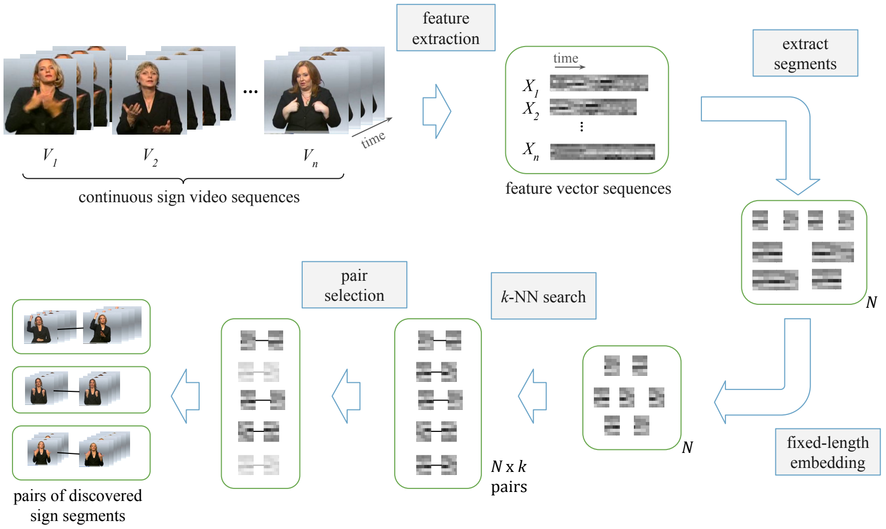


Fig. 1: Flow diagram of the KNN based discovery algorithm

2 K-Nearest Neighbours Based Discovery Algorithm

We adopt the KNN based discovery pipeline in [23], which begins with extracting large number of overlapping segments from the input sequences. These segments, which may have variable lengths, are transformed into fixed dimensional representations using smoothed sampling. Then for each segment representation, k nearest segments are searched so that, each segment is paired with k other segments. From these segment pairs, the ones that overlap and that have lower similarities are discarded. The remaining pairs are the discovered pairs. The flow diagram of this algorithm is displayed in Figure 1 and details of these steps are explained in the following sections.

2.1 Temporal Segmentation

For an input sequence, the points that are a frames apart are selected as candidate segmentation points. The segments are extracted for all possible combinations of these candidate points. As the parameter a decreases, the chance of finding correct boundaries increases at the expense of more computational cost. The segment lengths are constrained to an interval, which can be adjusted according to the expected term lengths. More formally, for a given d dimensional feature vector time series $X \in \mathbb{R}^{d \times T}$ of length T, a set of segments $\{X_{ij}\}$ are extracted such that

$$i, j \in \{0,\ a,\ 2a,\ \dots\ ,\lfloor T/a \rfloor \cdot a\} \tag{1}$$

$$l_{min} < j - i < l_{max} \tag{2}$$

where l_{min} and l_{max} set the bounds for segment lengths. This procedure considers all possible segments as candidates.

2.2 Fixed-Length Representations

We apply the embedding method described in [23], which simply is the sampling of input vectors, weighted by Gaussian kernels. A segment X_{ij} of L_0 frames is multiplied with a transformation matrix $F \in \mathbb{R}^{L_0 \times L}$ to be mapped to L-frame representation. The l^{th} column of F is the kernel defined as

$$f_l = \mathcal{N}\left(\frac{l \cdot L_0}{L}\ ,\ r \cdot L_0 + s \cdot g_L(l)\right) \tag{3}$$

where $g_L(l)$ is a triangular function such that $g_L(l) = \frac{L}{2} - \left|\frac{L}{2} - l\right|$, r and s are weighting parameters for the kernel's variance. The triangular function makes the frames in the middle more smoothed. This is a very simple method for obtaining fixed dimensional representation and more complex representation learning methods can be incorporated to this step.

2.3 Nearest Neighbour Search

Fixed-length representations are reshaped to 1d vectors so that each segment is represented by one of these vectors. The next step is to collect all of them to a search index, using the FAISS [5] framework, which builds a very efficient search index on GPU and can be scaled up easily. Then for each segment representation, the k nearest segments are found using Euclidean distance. If there are N segments, the search yields $N \times k$ pairs of similar segments.

2.4 Pair Selection

The pairs after the KNN search are mostly redundant because they overlap with each other. Therefore a series of elimination steps are required, so that only non-overlapping high confidence matches remain. The first step is to retrieve and sort all neighbours for an input file, and select only the top δ percent of the pairs. In other words, for an input file i, there are $N_i \times k$ pairs and we select the best $\delta \times N_i \times k$ pairs. The next step is to remove the self-overlapping pairs, whose segments overlap with each other. For a pair $p = (s_1, s_2)$, the self-overlap ratio between the segments s_1, s_2 is computed as the lengths of intersection over union

$$r_{self}(s_1, s_2) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|}. \tag{4}$$

The final step removes the remaining overlapping pairs by using non-maximal suppression (NMS). All pairs are sorted by decreasing similarity scores. Beginning from the top, the pairs are compared to worse pairs. Worse pairs are discarded if they overlap with a better pair, where the pair-overlap ratio is computed as

$$r_{pair}(p_i, p_j) = \frac{|s_{i,1} \cap s_{j,1}| \cdot |s_{i,2} \cap s_{j,2}|}{|s_{i,1}| \cdot |s_{i,2}|}. \tag{5}$$

The δ parameter is adjusted to control accuracy-coverage trade-off in the original work [23]. In our implementation, we fix $\delta = 5\%$ and apply another similarity threshold θ at the very end, in order to perform NMS only once and adjust the coverage after all computations are complete.

3 Setup for Sign Language Experiments

We test the term discovery pipeline described in Section 2 on sign language videos, by feeding visual features instead of acoustic features, using a similar setup to [20]. The visual features consist of hand shape and pose features which are obtained by running pre-trained models on each video frame. We use the Phoenix Weather 2014 [10] dataset, in which the gloss time boundaries are labelled, enabling us to evaluate the quality of discovered segments. Using the metrics for spoken UTD [15], we compare the KNN based algorithm [23] to the SDTW baseline [4] and comment on their differences.

3.1 Visual Features

Following the feature extraction steps in [20], we obtain two set of features by running pre-trained models on each video frame.

Hand Shapes. We use the DeepHand [11] pre-trained hand shape classifier network. It was originally trained over 1 million right hand images from three different sign corpora [2, 13, 10], where the hand shape labels were derived according to SignWriting notation [21]. For each video frame, we extract the 61 dimensional final layer activations before the softmax layer. We observed that reducing the dimensions to 40 by applying whitened PCA transformation improved the results in the discovery experiments. We only use the right hand features for all experiments.

Joint Locations. We also use the 2D joint coordinates that are found by running OpenPose [1] estimator on each frame. Concatenating the 8 upper body joints together with 21 keypoints for right and left hand each, we obtain 100 dimensional pose features per frame. The coordinates are normalized by subtracting the neck location and dividing by shoulder length.

3.2 Evaluation Criteria

There are numerous metrics for measuring different aspects of discovery systems. We base our metrics on the ones used in Zero Resource challenges [25, 3] which are described in detail by Ludusan et. al. [15]. We use the publicly available TDE toolkit[1] designed for spoken UTD. We modify some the metrics for our application.

The metrics are computed using the gold transcriptions of discovered segments. In spoken term discovery, the transcriptions are usually at phoneme level and a segment is associated with gold phones if the segment interval overlaps with more than 50% or 30ms of the phone duration. In our application, we only use the 50% criteria because the gold gloss lengths may vary significantly.

- Coverage: It is the ratio of non-overlapping discovered tokens to the discoverable tokens in all input sequences. In speech, it is computed as discovered phones over all phones; because all phones are assumed to be discoverable since there are usually less then a hundred phones for any spoken language. However, unlike speech, some sign types may appear only once in the whole input. Therefore, to be fair, we divide by the number of discoverable tokens, whose types are seen at least two times.

- Normalized Edit Distance (NED): It measures the quality of pairs, in terms of Levenshtein distance, which simply is the minimum number of modifications (insertion, deletion, substitution) required to make two discrete sequences the same, normalized by the length of the longer sequence. The final NED score is averaged over all pairs.

- Grouping Quality: This set of metrics is computed in terms of precision (P), recall (R) and F-score (harmonic mean of P, R). It is similar to cluster purity and inverse purity. If the pairs within a cluster has the same transcription, then the precision is high. If pairs from separate clusters have the same transcriptions, then the recall is low. For our application we don't expect grouping recall to be high since we don't perform further clustering step and leave them in pairs.

Type/token metrics analyse whether discovered groups of sub-units (phones) correctly represent the units. We don't report type/token metrics because available sign labels are not in the same granularity as phonemes in speech. The gloss labels we have correspond to words in speech. Therefore we report NED and grouping quality metrics, which are still significant even computed with gloss labels instead of sub-units.

[1] github.com/bootphon/tdev2

3.3 Dataset

The dataset we use is the Phoenix Weather 2014 [10], which consists of German Sign Language interpretations of weather forecast aired in a public TV. This dataset consists of 25 fps, RGB videos that are recorded in similar conditions where all signers face directly into the camera. We use the training set of multi-signer (MS) setup, which contains 5671 sentences that total 10 hours of videos. We use this subset because the gloss labels with time boundaries are provided only for this subset. These labels are automatically aligned by HMM-LSTM based model [12] using the sentence level gloss labels, annotated by human experts. The automatic frame level labels also indicate the HMM states of the HMM-LSTM based model [12] but we omit this information and use only the gloss information.

In addition to having time boundaries for labels, this corpus possesses other benefits for our task. The vocabulary is limited to weather related terms; there are only 1081 unique gloss types. Moreover, the signers are professionals which makes the inter-signer variability minimal.

Table 1: Partitions of the Phoenix 2014 MS [10] for cross-validated experiments

Subsets	Signer IDs	# Sentences	Total Size
1	4	836	1705
	8	704	
	9	165	
2	1	1475	1975
	3	470	
	6	30	
3	5	1296	1991
	7	646	
	2	49	

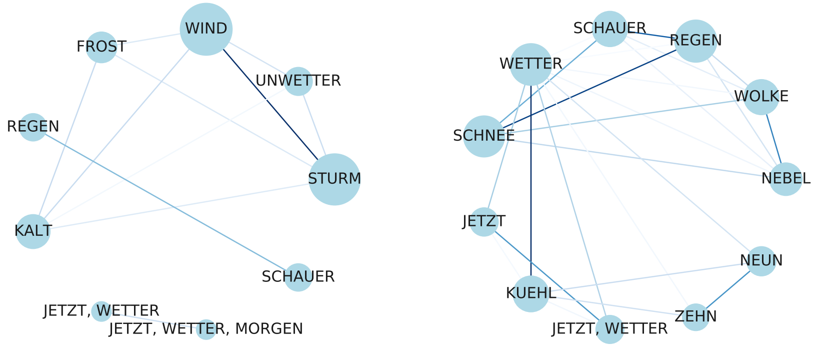
We partition the data into three folds for cross-validation, as shown in Table 1. We aim a partition where the subset sizes in terms of number of sentences are matched. At each fold, 1/3 of the data is used as development set and the remaining is used as unseen test set. Then we switch the development set and re-run the tuning procedure. The results are then reported using the average of test results, weighted by number of sentences. For each experiment, the final score threshold θ is adjusted so that Coverage is about 10%, and NED score is used as decision criterion.

4 Results and Discussion

In this section, we first discuss the KNN based algorithm [23] and then compare it to the SDTW baseline presented in [20].

4.1 KNN Based Discovery

We first explored the effect of different hyper-parameters on discovery performance. The expected term length for the Phoenix dataset [10] is about 10 frames (0.4s). Using this information, we set the minimum segment length l_{min} as 6 frames and segmentation resolution a to be 3 frames. We observed that setting the maximum segment length l_{max} as 45 frames (1.8s) allowed discovery of n-grams. With a, l_{min} and l_{max} fixed, we then perform cross-validated grid search to find the best combination of embedding dimension L and smoothing parameters r and s. Even though the optimum values for these parameters vary for each signer and type of feature, we observed that setting $r = 0.1, s = 0.4$ and $L = 6$ frames yield good results in general.



(a) KNN (DH)

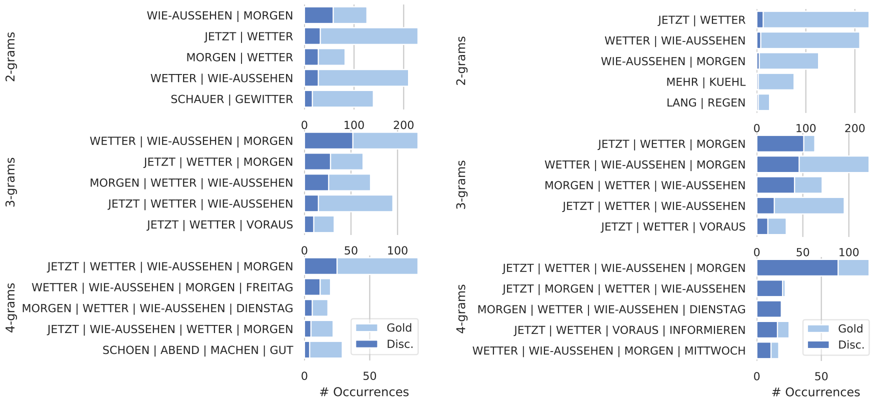
(b) KNN (OP)

Fig. 2: Examples of most confused pairs of glosses for hand shape (DH) and pose (OP) features. Darker lines represent more confused pairs and circle radii are proportional to gloss frequencies

Among the two sets of features (DeepHand and OpenPose [11, 1]), the DeepHand features yield better results as shown in Table 2. The most confused pairs for each type of feature are given in Figure 2. Here, we observe that semantically similar signs (e.g. rain-shower, wind-storm etc.) which also have similar forms are easily confused. Interestingly, the clusters of most confused glosses differ according to feature type. This observation leads to the conclusion that in future studies, these two types of features can be fused together to complement each others weaknesses.

4.2 Comparison to SDTW Baseline

The segmental DTW based discovery algorithm [4] is regarded as the baseline for Zero Resource speech challenges [25, 3]. It is also the only algorithm previously adapted for sign terms discovery [20]. Therefore we use it as the baseline algorithm to compare our results. We rerun this algorithm in order to obtain the matching pairs without clustering step, so that only matching stages are compared. Since this algorithm is reported to work poorly with pose features [20], we run with the hand shape features only. We apply a similar procedure for removing overlapping pairs, as described in Section 2.4.



(a) KNN (DH)

(b) SDTW (DH)

Fig. 3: Examples of correctly discovered gloss n-grams (Disc.) for both algorithms, together with the number of occurrences of existing gold n-grams

The biggest difference between the KNN based algorithm [23] and SDTW baseline [4] is the length of discovered segments (see Table 2). The SDTW baseline is able to discover longer segments, often n-grams, therefore less pairs are needed to satisfy 10% Coverage. Correctly discovered n-grams for both algorithms are displayed in Figure 3. The baseline algorithm uses a sparse approximation of the distance matrix. Then the SDTW search is performed on the diagonal segments that remain after the median filtering of the sparse matrix. Median filtering allows only long diagonal paths to be searched therefore the segments with this method tend to be longer. Conversely, KNN based algorithm is more receptive to shorter segments, because smoothed embedding method may cause less distortion for shorter segments.

Table 2: Term discovery results of both algorithms at 10% Coverage, averaged for unseen test sets

	NED	Grouping			Avg. Seg.	# Discovered
Experiment	(%)	P (%)	R (%)	F (%)	Length (sec)	Pairs
SDTW (DH)	**41.0**	18.9	51.8	27.4	2.1	994.9
KNN (OP)	50.7	43.1	39.7	41.0	0.4	1206.4
KNN (DH)	43.4	**50.1**	**52.0**	**51.0**	0.5	1359.2

As shown in Table 2, NED scores for both algorithms are similar. However, grouping precision of the KNN based algorithm is considerably better. This is because the grouping quality metrics are originally designed for evaluating clusters that have more than two segments. Since we don't perform clustering, each cluster has exactly two segments, and therefore the grouping precision gives the ratio of perfect matches over all pairs. As a result, the partial matches between longer segments do not count as positive examples. Another notable difference is that, using pose features does not significantly degrade discovery performance for KNN based algorithm, whereas it was reported to degrade performance in [20]. It should also be noted that KNN based algorithm runs much faster; precomputed features for 3 hours of video is processed in about one minute using GPU, versus 10 minutes using the SDTW baseline.

For all setups, nearly 1% of the perfect matches come from different signers, most of the correct matches belong to the same signer. Therefore we can think of these results as the average of signer dependent experiments.

5 Conclusions

In this work, we demonstrate that a KNN based spoken term discovery algorithm [23] can be run for continuous sign language to discover sign terms, by using features extracted from RGB videos only. We compare this algorithm to the baseline SDTW method proposed in [20], using the same dataset [10] and similar metrics. We show that the baseline method is better at discovering longer sequences and KNN method is better for discovering shorter segments. Nonetheless, the KNN based method runs much faster. It is also more flexible in the sense that, more sophisticated segmentation and embedding approaches can be incorporated easily. Henceforth, a future direction is to focus on representation learning methods [14], which may also combine non-manual modalities. Using the discovered pairs, representation learning methods such as frame-wise correspondence autoencoders (CAE) [7] or sequence to sequence CAE [6] can be used. The cross-validated evaluation scheme that we propose may be used in future studies to benchmark other UTD algorithms and representation learning methods.

6 Acknowledgments

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

References

1. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: Proc CVPR. pp. 1302–1310 (2017)
2. D. McKee, R. McKee, S. P. Alexander, and L. Pivac: The online dictionary of new zealand sign language (2015), http://nzsl.vuw.ac.nz
3. Dunbar, E., Cao, X., Benjumea, J., Karadayi, J., Bernard, M., Besacier, L., Anguera, X., Dupoux, E.: The zero resource speech challenge 2017. In: Proc. ASRU. pp. 323–330 (Dec 2017)
4. Jansen, A., Durme, B.V.: Efficient spoken term discovery using randomized algorithms. In: Proc. ASRU (2011)
5. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 (2017)
6. Kamper, H.: Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6535–3539 (May 2019)
7. Kamper, H., Elsner, M., Jansen, A., Goldwater, S.: Unsupervised neural network based feature extraction using weak top-down constraints. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5818–5822 (April 2015)
8. Kamper, H., Jansen, A., Goldwater, S.: Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. IEEE/ACM Transactions on Audio, Speech, and Language Processing **24**(4), 669–679 (April 2016)
9. Kelly, D., Mc Donald, J., Markham, C.: Weakly supervised training of a sign language recognition system using multiple instance learning density matrices. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **41**(2), 526–541 (April 2011)
10. Koller, O., Forster, J., Ney, H.: Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding **141**, 108–125 (Dec 2015)
11. Koller, O., Ney, H., Bowden, R.: Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In: Proc. CVPR. pp. 3793–3802 (Jun 2016)
12. Koller, O., Zargaran, S., Ney, H.: Re-Sign: Re-Aligned End-to-End Sequence Modelling with Deep Recurrent CNN-HMMs. In: Proc. CVPR. pp. 3416–3424 (Jul 2017)

13. Kristoffersen, J. H., T. Troelsgård, A. S. Hardell, B. Hardell, J. B. Niemelä, J. Sandholt, and M. Toft: Ordbog over dansk tegnsprog (2008-2016), http://www.tegnsprog.dk
14. Levin, K., Henry, K., Jansen, A., Livescu, K.: Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding pp. 410–415 (2013)
15. Ludusan, B., Versteegh, M., Jansen, A., Gravier, G., Cao, X.N., Johnson, M., Dupoux, E.: Bridging the gap between speech technology and natural language processing: an evaluation toolbox for term discovery systems. In: Proc. Language Resources and Evaluation Conference (May 2014), https://hal.inria.fr/hal-01026368
16. Nayak, S., Duncan, K., Sarkar, S., Loeding, B.L.: Finding recurrent patterns from continuous sign language sentences for automated extraction of signs. Journal of Machine Learning Research **13**, 2589–2615 (2012)
17. Ondel, L., Burget, L., Černocký, J.: Variational inference for acoustic unit discovery. Procedia Computer Science **81**, 80–86 (12 2016)
18. Park, A.S., Glass, J.R.: Unsupervised pattern discovery in speech. IEEE Transactions on Audio, Speech, and Language Processing **16**(1), 186–197 (Jan 2008)
19. Pfister, T., Charles, J., Zisserman, A.: Large-scale learning of sign language by watching TV (using co-occurrences). Proceedings of the British Machine Vision Conference pp. 1–11 (2013)
20. Polat, K., Saraçlar, M.: Unsupervised term discovery for continuous sign language. In: Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives. pp. 189–196. European Language Resources Association (ELRA), Marseille, France (May 2020)
21. Sutton, V.: Sign writing. Deaf Action Committee (DAC) for Sign Writing (2000)
22. Theodorakis, S., Pitsikalis, V., Maragos, P.: Dynamic-static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition. Image Vision Comput. **32**, 533–549 (2014)
23. Thual, A., Dancette, C., Karadayi, J., Benjumea, J., Dupoux, E.: A K-nearest neighbours approach to unsupervised spoken term discovery. In: IEEE Spoken Language Technology SLT-2018. Proceedings of SLT 2018, Athènes, Greece (Dec 2018)
24. Tornay, S., Magimai.-Doss, M.: Subunits inference and lexicon development based on pairwise comparison of utterances and signs. Information **10**, 298 (2019)
25. Versteegh, M., Thiollière, R., Schatz, T., Cao Kam, X.N., Anguera, X., Jansen, A., Dupoux, E.: The zero resource speech challenge 2015. In: Proc. Interspeech. pp. 3169–3173 (2015)
26. Yin, P., Starner, T., Hamilton, H., Essa, I., Rehg, J.M.: Learning the basic units in american sign language using discriminative segmental feature selection. In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 4757–4760 (2009)