Listen, Decipher and Sign: Toward Unsupervised Speech-to-Sign Language Recognition

Liming Wang[1], **Junrui Ni**[1], **Heting Gao**[1], **Jialu Li**[1], **Kai Chieh Chang**[1], **Xulin Fan**[1], **Junkai Wu**[1], **Mark Hasegawa-Johnson**[1] and **Chang D. Yoo**[2]
[1]University of Illinois Urbana-Champaign
[2]Korea Advanced Institute of Science Technology
{lwang114,jhasegaw}@illinois.edu, cd_yoo@kaist.ac.kr

Abstract

Existing supervised sign language recognition systems rely on an abundance of well-annotated data. Instead, an unsupervised speech-to-sign language recognition (SSR-U) system learns to translate between spoken and sign languages by observing only non-parallel speech and sign-language corpora. We propose speech2sign-U, a neural network-based approach capable of both character-level and word-level SSR-U. Our approach significantly outperforms baselines directly adapted from unsupervised speech recognition (ASR-U) models by as much as 50% recall@10 on several challenging American sign language corpora with various levels of sample sizes, vocabulary sizes, and audio and visual variability. The code is available at cactuswiththoughts/UnsupSpeech2Sign.git.

1 Introduction

Many hearing-impaired people communicate natively in sign language (SL); for them, SL communication is as effortless as native spoken communication is for normal-hearing people. However, when it comes to a conversation between a hearing-impaired and a normal hearing, tremendous barriers exist for several reasons. First, there is a shortage of people who are bilingual in spoken and sign languages. Automatic sign language recognition models exists (Koller et al., 2016; Huang et al., 2018) but are fully supervised and require a large number of annotated data, which are hard to acquire. As a result, such systems are often limited to a small vocabulary. On the other hand, untranscribed speech audio and SL videos are quite common on the Internet, presenting an exciting possibility: Given a non-parallel pair of speech and sign language datasets, can we train a model to translate between spoken and sign languages? This task, we called *unsupervised speech-to-sign language recognition* (SSR-U), is analogous to well-known problems such as unsupervised machine translation (MT-U) (Ravi and Knight, 2011; Artetxe et al., 2018a; Lample et al., 2018) and unsupervised automatic speech recognition (ASR-U) (Liu et al., 2018; Chen et al., 2019; Baevski et al., 2021), albeit with a few new challenges. First of all, in the case of SSR-U, both modalities are continuous as opposed to at least one of them being discrete in the case of ASR-U and MT-U. Consequently, the matching process is much more challenging due to higher within and cross-modal variability. Further, most sign language and spoken language can only be matched on the *word* level as opposed to the subword level in the case of ASR-U. Not only does the space of possible mappings explode combinatorially, but less training data and fewer temporal constraints are also available to recover the correct mapping.

In this paper, we develop a neural network-based framework, speech2sign-U, for both character-level (with fingerspelling sequence) and word-level SSR-U. It achieves promising results on datasets with up to around 900 ASL signs.

2 Problem formulation

Suppose we have a corpus of unlabeled speech recordings sampled from the random process $A = (A_1, \cdots, A_T)$ and another separately collected corpus of unlabeled sign language videos sampled from the random process $V = (V_1, \cdots, V_L)$. Both A and V contain the *same* semantic information but different para-linguistic information such as speaker/signer identity and prosody. In other words, if we filter out the para-linguistic information and retain the semantic information as $X := X(A) = (X_1, \cdots, X_T) \sim P_X$ for the speech and $Y := Y(V) = (Y_1, \cdots, Y_L) \sim P_Y$ for the videos, we can find a *generator* function $\mathcal{G} : \mathbb{X}^T \mapsto \mathbb{Y}^L$ such that $Y = \mathcal{G}(X)$. Since the corpora are unpaired, we cannot estimate $\mathcal{G}$ directly from samples, and the goal of SSR-U is to "decipher" it using only the relations between the speech-only and video-only distributions, P_X and

Findings of the Association for Computational Linguistics: ACL 2023, pages 6785–6800
July 9-14, 2023

P_Y:

$$\sum_{x \in \mathbb{X}^T} P_X(x) G(y|x) = P_Y(y), \quad (1)$$

for all sign language unit sequences $y \in \mathbb{Y}^L$, where $G(y|x) = 1$ if and only if $y = \mathcal{G}(x)$.

3 Proposed Methods

3.1 Character-level speech2sign-U

In the case of character-level speech2sign-U, V is drawn from a collection of unlabeled fingerspelling sequences, where each V_i is the hand gesture for a character. In this case, we adopt a similar architecture as wav2vec-U (Baevski et al., 2021).

Sign video preprocessing Given a sign video $v \sim V$, we obtain its visual features $(v_1, \cdots, v_L)$ by passing the raw video frames into a *local* feature extractor such as VGG19 or RCNN (Ren et al., 2015). The local features are then contextualized by a *sign language encoder*, consisting of a two-layer multilayer perceptron (MLP) and a one-layer uni-directional LSTM:

$$c_1, \cdots, c_L = \texttt{LSTM}(v_1, \cdots, v_L). \quad (2)$$

The sign language encoder is then trained using contrastive predictive coding (CPC) (van den Oord et al., 2018):

$$\mathcal{L}_{\text{CPC}} := - \mathbb{E}_V \left[\sum_{i,k} \log \frac{e^{c_i^\top \texttt{MLP}(v_{i+k})}}{\sum_{n \in \mathcal{N}_{i,k}} e^{c_i^\top \texttt{MLP}(n)}} \right], \quad (3)$$

where $\mathcal{N}_{i,k}$ is a set of negative samples chosen uniformly at random from times other than $i + k$. Finally, we apply K-means clustering on $(c_1, \cdots, c_L)$ to obtain the *sign cluster units* $Y := (y_1, \cdots, y_L)$.

Speech preprocessing As in wav2vec-U, for each utterance, we first use a voice activity detector (VAD) to remove silences between speech frames and randomly insert silences between word boundaries of the sign cluster sequence so that their silence distributions match. Next, we contextualize the raw speech frames using wav2vec 2.0 pretrained on LibriLight:

$$(z_1, \cdots, z_T) = \texttt{wav2vec2}(a_1, \cdots, a_T). \quad (4)$$

Finally, we extract K-means clusters from $(z_1, \cdots, z_T)$ and merge consecutive frames belonging to the same clusters to obtain the segment-level speech features $(x_1, \cdots, x_T)$.

Unsupervised training A convolutional generator $G : \mathbb{X} \to \mathbb{Y}$ then generates a sequence of cluster units $(\hat{Y}_1, \cdots, \hat{Y}_L) = \mathcal{G}(X)$ from the segment features X by sampling from the posterior probabilities at each segment i:

$$\hat{Y}_i \sim G_i(y_i|X) := \frac{\exp(\texttt{Conv}_{i,y_i}(X))}{\sum_k \exp(\texttt{Conv}_{i,k}(X))}. \quad (5)$$

Then we adopt the generative adversarial network (GAN (Goodfellow et al., 2014)) objective by training a binary classifier $D : \mathbb{Y} \mapsto [0, 1]$ to discriminate between the real cluster sequence and the generated one:

$$\begin{aligned} \min_D \max_G & -\mathbb{E}_{X \sim P_X} \left[\log(1 - D(\mathcal{G}(X))) \right] \\ & - \mathbb{E}_{Y \sim P_Y} \left[\log D(Y) \right] \\ & + \lambda \mathcal{L}_{\text{gp}} + \gamma \mathcal{L}_{\text{sp}} + \eta \mathcal{L}_{\text{cd}}, \end{aligned} \quad (6)$$

where $\mathcal{L}_{\text{gp}}, \mathcal{L}_{\text{sp}}$ and $\mathcal{L}_{\text{cd}}$ stand for the gradient penalty, smoothness penalty and code diversity losses as defined in (Baevski et al., 2021).

3.2 Word-level speech2sign-U

Word-level speech2sign-U is more challenging than character-level: the GAN objective in Eq. (6) fails to converge for vocabulary sizes of 100 or larger, apparently due to variability in the audio and video signals. Therefore, we instead adopt a novel GAN-*free* architecture trained to match marginals between the generated and real probability distributions as shown in Fig. 1.

Preprocessing We extract the sign video and speech features similar to Section 3.1, except with a few modifications: first, we assume the word-level boundaries for both the speech and sign videos are available, which may be ground truth or boundaries detected using unsupervised word segmentation algorithms from phoneme boundaries (Kreuk et al., 2020; Bhati et al., 2021; Cuervo et al., 2022). Then we compute the segment-level speech features by averaging the frame-level wav2vec 2.0 features within each word. Further, we use the I3D (Carreira and Zisserman, 2017) as the local feature extractor and average the pretrained video feature frames within each word-level sign video segment. Lastly, we perform K-means clustering on the segment features and use the output cluster units as inputs X to the speech generator as we found that quantized speech features work better than continuous features.

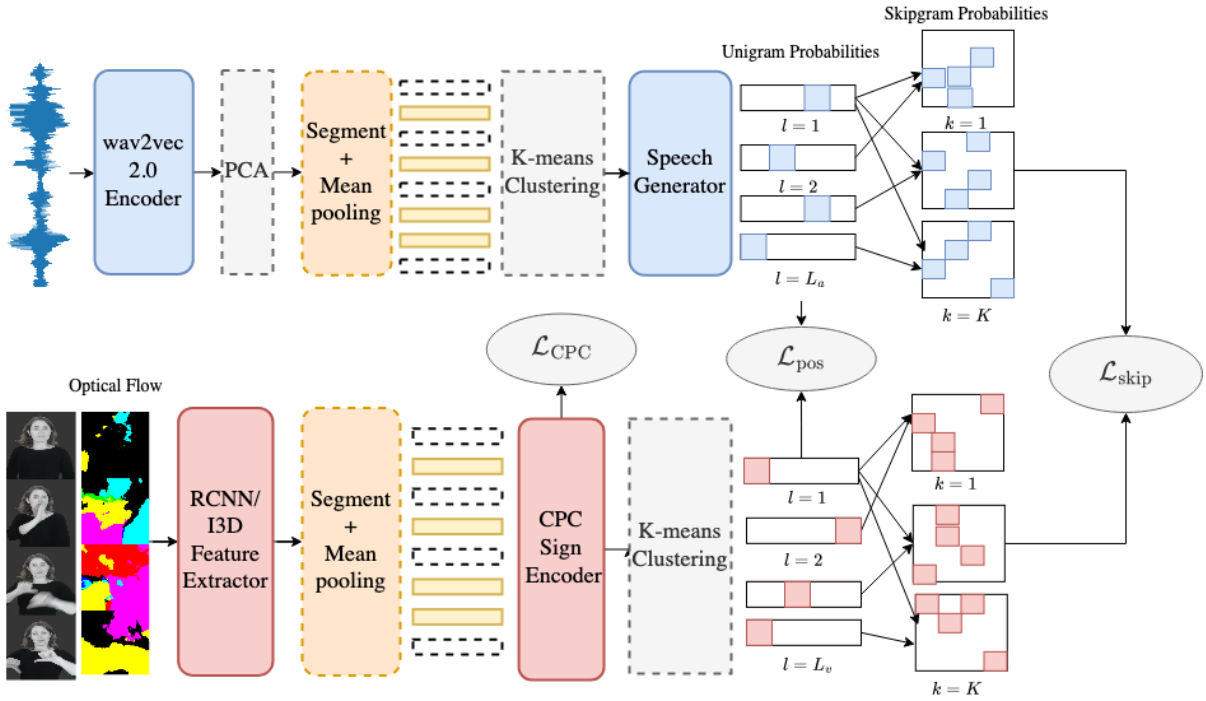


Figure 1: Overall architecture of the word-level speech2sign-U. Solid blocks contain trainable parameters while dashed blocks do not.

Unsupervised unigram matching Similar to Section 3.1, we seek to match the probability distributions in the two modalities as our unsupervised training criterion. Instead of using a convolutional generator as in Eq. (5), we instead use a *linear* generator for each segment i:

$$G(y_i|x_i) := \frac{\exp(W_{y_i}x_i)}{\sum_{y'\in\mathbb{Y}}\exp(W_{y'}x_i)}, \tag{7}$$

Eq. (1) can now be achieved by minimizing the ℓ_1 distance between the empirical *positional unigram* probabilities of the generated and real sign cluster units:

$$\mathcal{L}_{\text{pos}}(G) = \sum_{i=1}^{L} \|\hat{P}_{X_i}G - \hat{P}_{Y_i}\|_1, \tag{8}$$

where $\hat{P}_{X_i}$ and $\hat{P}_{Y_i}$ are empirical unigram distributions for the speech and sign units, and $G \in \mathbb{R}^{|\mathbb{X}|\times|\mathbb{Y}|} := (G(y|x))_{x\in\mathbb{X},y\in\mathbb{Y}}$. Note that such an objective is typically optimized implicitly by a GAN, but we found that the explicit formula not only avoids the need for a discriminator but also leads to more stable training and better performance.

Unsupervised skip-gram matching Positional unigram constraints alone may not be sufficient for word-level SSR-U. Therefore, we add additional moment constraints using *skip-grams*. Define the k-step skip-gram to be the joint probability

$$\begin{aligned}\Pr[Z_1 = z, Z_{k+1} = z'] &:= \frac{\sum_{i=1}^{L-k} P_{Z_iZ_{i+k}}(z,z')}{L-k} \\ &=: (P_k^{ZZ'})_{zz'}. \end{aligned} \tag{9}$$

Then, apply Eq. (1) again, we have the skip-grams for the generated and real sign cluster units satisfy

$$G^\top P_k^{XX'}G = P_k^{YY'},\ 1 \le k \le K-1. \tag{10}$$

Again, we approximate this constraint by minimizing their ℓ_1 distance:

$$\mathcal{L}_{\text{skip}}(G) = \sum_{k=1}^{K} \|G^\top \hat{P}_k^{XX'}G - \hat{P}_k^{YY'}\|_1. \tag{11}$$

The overall loss for the word-level speech2sign-U is then

$$\mathcal{L}_{\text{sp2sign-U,word}} = \mathcal{L}_{\text{pos}} + \lambda\mathcal{L}_{\text{skip}}. \tag{12}$$

Speech-to-sign retriever Given a query speech audio (sign video), we would like to use it to retrieve its translation from a database of sign videos (speech audios). To this end, we use the generator

	# signs	# train sents	# valid	# test
Character-level datasets				
FS LibriSpeech	87k	287k	5.5k	5.6k
FS LJSpeech	87k	13.1k	348	523
Word-level datasets				
ASL Libri. 100	2.6k	14.1k	56	54
ASL Libri. 200	4.4k	56.2k	291	311
ASL Libri. 500	8.2k	137k	941	956
ASL Libri. 1k	11.6k	290k	2.7k	2.5k

Table 1: Dataset statistics

to compute a similarity score between each speech sequence X and sign sequence Y as:

$$\text{Sim}(X, Y) = -\frac{1}{L}\texttt{DTW}(G(X), Y), \qquad (13)$$

where $\texttt{DTW}(\cdot, \cdot)$ is the dynamic time warping distance between two feature sequences with *cosine distance* as the frame-level metric, computed using the DTW library (Giorgino, 2009).

4 Experiments

4.1 Datasets

The detailed statistics are shown in Table 1.

Fingerspelling LibriSpeech To extract semantic units from the fingerspelling signs, we trained the visual CPC encoder on a sentence-level fingerspelling dataset constructed from the 960-hour LibriSpeech dataset and the Unvoiced dataset (Nagaraj, 2018). To construct the dataset, we replace each letter in the LibriSpeech transcript with an image of that letter's ASL Alphabet symbol chosen uniformly at random from Unvoiced. To study the effect of visual variability on SSR-U, we subset the ASL Alphabet images to 100, 300, 500, or 1000 images per letter sign. The dev-clean subset of LibriSpeech is used as the validation set.

Fingerspelling LJSpeech We train our character-level model on another sentence-level fingerspelling dataset constructed from LJSpeech (Ito and Johnson, 2017) and the ASL Alphabet dataset similar to the fingerspelling LibriSpeech.

ASL LibriSpeech For the word-level SSR-U, we construct another corpus using LibriSpeech for speech and MSASL (Joze and Koller, 2019) for word-level sign videos. Since many MSASL videos no longer exist on YouTube, only 11.6k out of 25k videos are downloaded. Further, due to the mismatch in vocabulary size, we use forced alignment information to filter out LibriSpeech words that don't appear in MSASL and keep sentences that are at least 5 words long. Next, for each word in each sentence, we pick a word-level sign video uniformly at random from MSASL. To study the effect of vocabulary size on our model, we follow the split provided by (Joze and Koller, 2019) to subset the data to a vocabulary size of 100, 200, 500 or 1000.

4.2 Overall results

Evaluation metrics We evaluate the performance of our systems using two metrics: the *unit error rate* (UER) is the average insertion I, deletion D, and substitution S error between the predicted and true visual cluster units, which may be character- or word-level units depending on the task:

$$\text{UER} = \frac{I + D + S}{3} \times 100.$$

The other metric we used to evaluate the speech-to-sign (A$\rightarrow$ V) and sign-to-speech (V$\rightarrow$ A) retrieval tasks is *recall@k* (R@k) ($k = 1, 5, 10$), which is the percentage of hits in the top k results returned by the retriever.

Character-level SSR-U The character-level results are shown in Table 2. To obtain retrieval results, we trained our own wav2vec-U 2.0 using the code released by the authors. Unfortunately, we were unable to achieve the same results they report in their paper. For our ASR-U experiments, wav2vec-U significantly outperforms wav2vec-U 2.0 in terms of both word error rates and retrieval tasks. For SSR-U, we compare our models with wav2vec-U (and 2.0) as well as a supervised image and caption retrieval model trained under a ranking-based criterion (Harwath et al., 2018). We replace their original CNN speech encoder with a two-layer MLP with hidden and output sizes of 256 and ReLU activation, and their VGG16 image encoder with a linear image encoder with an output size of 256. We found that our models with 100 and 300 images per letter achieve superior performances in terms of recall scores, even to the text-based wav2vec-U, but remain about 30% below the supervised topline. Notably, our model performs worse on the $A \rightarrow V$ direction than on the $V \rightarrow A$ direction, especially in terms of recall@1. This is perhaps due to significant insertion

Model	Images/ltr	UER↓	A→ V			V→ A		
			R@1↑	R@5↑	R@10↑	R@1↑	R@5↑	R@10↑
Supervised speech-to-sign recognition								
(Harwath et al., 2018)	1000	-	85.1	93.7	95.4	79.3	96.8	99.1
Unsupervised speech recognition								
wav2vec-U	-	**39.5**	**1.9**	**42.1**	**59.5**	**17.6**	**44.2**	**65.2**
wav2vec-U 2.0	-	68.1	1.1	6.3	11.8	2.6	9.8	14.9
Unsupervised speech-to-sign recognition								
speech2sign-U	100	**43.1**	1.7	42.1	63.4	27.5	**62.7**	**78.4**
speech2sign-U	300	45.0	**1.9**	**48.8**	**67.1**	22.8	53.9	71.5
speech2sign-U	500	46.2	1.0	33.1	57.2	31.3	57.0	72.8
speech2sign-U	1000	48.6	1.3	43.8	63.8	**32.5**	58.7	71.5

Table 2: Overall speech2sign-U results on FS LJSpeech

errors in the generated character sequence, which leads to many false positives during speech-to-sign retrieval.

Word-level SSR-U The word-level results are shown in Table 3. To establish top-line results for error rates and retrieval recall scores, we train a word-level unsupervised speech recognition model, speech2text-U, using the same criterion as speech2sign-U in Eq. 12, except by replacing the sign cluster sequences obtained from clustering word-level sign video features (see Section 3.2) with the underlying textual word labels as the target random variable Y. At the same time, for the subset with a vocabulary size of 98, we compare the performance of our model that uses unsupervised unigram and skipgram matching with wav2vec-U, which uses a JSD GAN for distribution matching, to show our proposed training method significantly improves the word error rates and the recall scores for both retrieval directions. However, we still observe a large gap in recall between our unsupervised model and the supervised speech-to-image retrieval model (Harwath et al., 2018). The performance of both word-level ASR-U and SSR-U degrades as the vocabulary size increases. The unit error rate (UER) increases from 53.6% to 87.9%, the recall@1 of speech-to-sign (A→V) retrieval decreases from 69.6% to 12.1%, and the recall@1 of sign-to-speech (V→A) retrieval decreases from 71.4% to 10.9% as the vocabulary size increases from 98 to 877. Such performance degradation is much more significant than that of character-level SSR-U because the word modality involves extra morphological complexity on top of the phonological character modality.

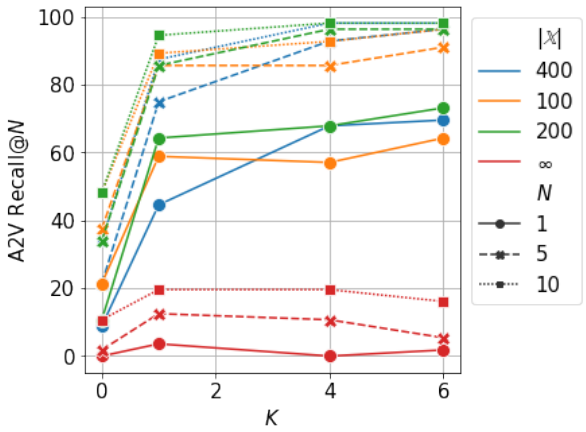


Figure 2: speech2sign-U retrieval results (recall@N, $N \in \{1, 5, 10\}$) vs skip-gram size K for various number of speech clusters on ASL LibriSpeech 100

4.3 Analysis

Effect of skip-gram size The relation between recall@1, 5, 10 and skip-gram size K is shown in Figure 2. Increasing K generally improves all recall metrics for SSR-U by introducing more constraints to the generator mapping, though the performance starts to saturate at $K = 4$.

Effect of the number of speech clusters We experiment with speech2sign-U models with speech cluster sizes $|\mathbb{X}|$ equal to 100, 200, 400, and a model that directly takes raw wav2vec 2.0 features as inputs ($|\mathbb{X}| = \infty$), as shown in Figure 2. We found that the continuous model is significantly

Model	Vocab size	UER↓	A→ V			V→ A		
			R@1↑	R@5↑	R@10↑	R@1↑	R@5↑	R@10↑
Supervised speech-to-sign recognition								
(Harwath et al., 2018)	877	-	55.2	78.9	86.3	51.7	85.5	93.1
Unsupervised speech recognition								
wav2vec-U	98	73.7	16.1	32.1	51.8	17.8	41.1	50.0
speech2text-U	98	**7.5**	**98.2**	**98.2**	**100**	**98.2**	**98.2**	**98.2**
speech2text-U	193	11.2	96.9	98.6	99.0	96.9	99.3	99.3
speech2text-U	468	30.0	68.0	86.2	90.2	66.7	85.3	90.2
speech2text-U	877	34.4	37.9	60.7	69.3	38.7	59.4	68.3
Unsupervised speech-to-sign recognition								
speech2sign-U	98	**53.6**	**69.6**	**96.4**	**98.2**	**71.4**	**96.4**	**100**
speech2sign-U	193	60.8	75.3	91.1	92.4	69.4	90.0	93.5
speech2sign-U	468	73.2	56.5	76.8	83.6	47.6	74.5	83.5
speech2sign-U	877	87.9	12.1	25.5	32.6	10.9	22.1	29.7

Table 3: Overall speech2sign-U results on ASL LibriSpeech

Video features	Vocab Size	A→ V		
		R@1	R@5	R@10
VGG19	98	32.1	64.3	78.6
OpenPose	98	0.0	8.9	16.1
I3D RGB	98	**76.8**	89.3	96.4
	193	66.0	86.3	91.4
	877	1.1	3.0	5.1
I3D flow	98	69.6	**96.4**	**98.2**
	193	63.9	86.9	91.1
	468	43.9	68.5	76.6
	877	**12.1**	**25.5**	**32.6**
I3D joint	98	28.6	67.9	76.8
	193	**75.3**	**91.1**	**92.4**
	468	**56.5**	**76.8**	**83.6**
	877	0.1	0.2	0.4

Table 4: Effect of the video features on ASL LibriSpeech with various vocabulary sizes

Boundary Label	Word Boundary F1	A→ V		
		R@1	R@5	R@10
speech2text-U				
Word	100	98.2	98.2	100
Phoneme	88.1	78.6	98.2	98.2
speech2sign-U				
Word	100	69.6	96.4	98.2
Phoneme	88.1	57.1	82.1	91.1

Table 5: Effect of the speech segmentation using speech2sign-U on ASL LibriSpeech 100

worse than discrete models and $|\mathbb{X}| = 200$ provides the most consistent recall scores across different skip-gram sizes.

Effect of training objectives The effect of different training objectives including the default speech2sign-U loss (L1) in Eq. (12), the maximum mean discrepancy (MMD) GAN and the Jensen-Shannon divergence (JSD) GAN is shown in Figure 3. For models trained with MMD and JSD GAN loss, we instead feed the generator outputs to a discriminator with a single convolutional layer while keeping all other settings the same. Our experiment indicates that the GAN-free approach is consistently more stable and accurate compared to the GAN-based approach.

Effect of visual features The effect of visual features is shown in Table 4. We experimented with different types of visual features on ASL LibriSpeech with different vocabulary sizes such as VGG19 and the pose keypoint features from OpenPose (Cao et al., 2019). For the OpenPose features, we extract the keypoints from each video frames and re-sample each sign video feature frames to 30 frames as the segment-level feature. I3D architecture (Carreira and Zisserman, 2017) significantly

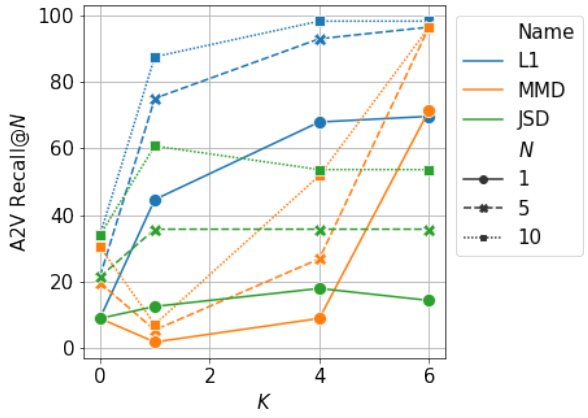


Figure 3: Retrieval results (recall@N, $N \in \{1, 5, 10\}$) vs skip-gram size K for various types of training objective on ASL LibriSpeech 100

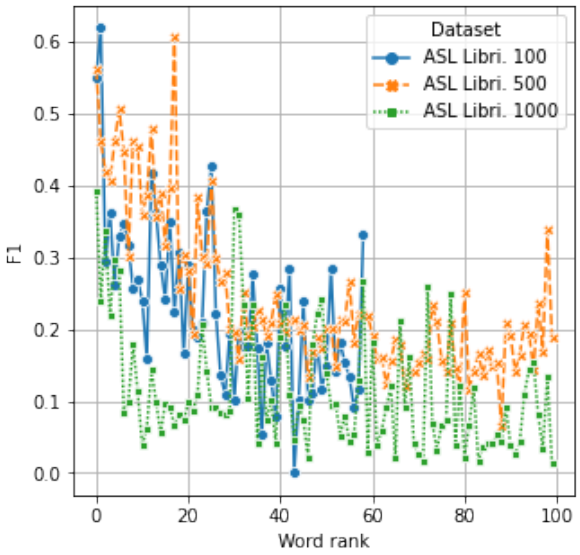


Figure 4: Word detection F1 vs. word rank by frequency

outperforms VGG19 and OpenPose as a feature extractor, demonstrating the importance of temporal information for SSR-U. We also found that I3D with optical flow features performs better than I3D with raw RGB inputs for most vocabulary sizes. Further, we found that concatenating the features from the RGB-based and flow-based I3Ds is beneficial for vocabulary sizes 193 and 468 but not when the vocabulary size is too small or too large, even causing training instability for vocabulary size 877.

Effect of segmentations The effect of gold and predicted speech segmentation for word-level SSR-U is shown in Table 5. For models trained with phoneme boundaries, we obtain predicted word segmentations using a CPC-based unsupervised segmentation system (Kreuk et al., 2020) with mean-pooled phoneme-level wav2vec 2.0 features as inputs. The convolutional encoder in the original model is replaced by a two-layer MLP with 256 output dimensions trained on ASL LibriSpeech 100 for 200 epochs. This yields an exact-match boundary F1 of 88%. Using such detected word boundaries, we found about a 20% drop in recall@1 for speech2text-U and an 8-17% relative drop in recall@1,5,10 for speech2sign-U. Still, our model remains much better than the wav2vec-U baseline with ground-truth word boundaries, demonstrating its robustness to segmentation noise.

Effect of word frequencies We plotted the F1 score of the first 100 word classes ranked by frequency in Figure 4. For ASL LibriSpeech 100 and 500, while noisy, it is not hard to observe that the F1 score positively correlates with word frequency in a somewhat exponential fashion. Starting with F1 above 0.55 for the most frequent word, the performance quickly drops below 0.2 at around the 30th most frequent word. This trend is less conclusive on ASL LibriSpeech 1000 with generally low F1 scores, but the highest F1 scores are still observed for the most frequent words. The trend is also illustrated by the DTW alignment of a speech-video pair correctly retrieved by speech2sign-U in Figure 5. In our example, speech2sign-U mistakes the sign "more" for more frequent signs such as "when" and "have". Additional factors such as *visual similarity* also play a role in the case of "more" and "when", as both signs involve touching the tips of both hands. Such factors may explain the fluctuations in Figure 4. More error analysis can be found in Appendix A.

5 Related works

Sign language recognition One way to bridge between sign language and written/spoken language is to build a sign language recognition (SLR) system trained on parallel sign language and text corpora. The earliest attempts tried to recognize fingerspelling gestures using hand-tracking signals from wired gloves (Grimes, 1983; Charayaphan and Marble, 1992). Later works introduced vision to either correct the errors made by the hand-tracking model, or to serve as a cheaper and less-intrusive alternative (Tamura and Kawasaki, 1988). Focusing on the problem of *isolated* sign recognition and treating it as a classification task, a variety of statistical and deep learning models have been proposed, such as HMM (Starner and Pent-

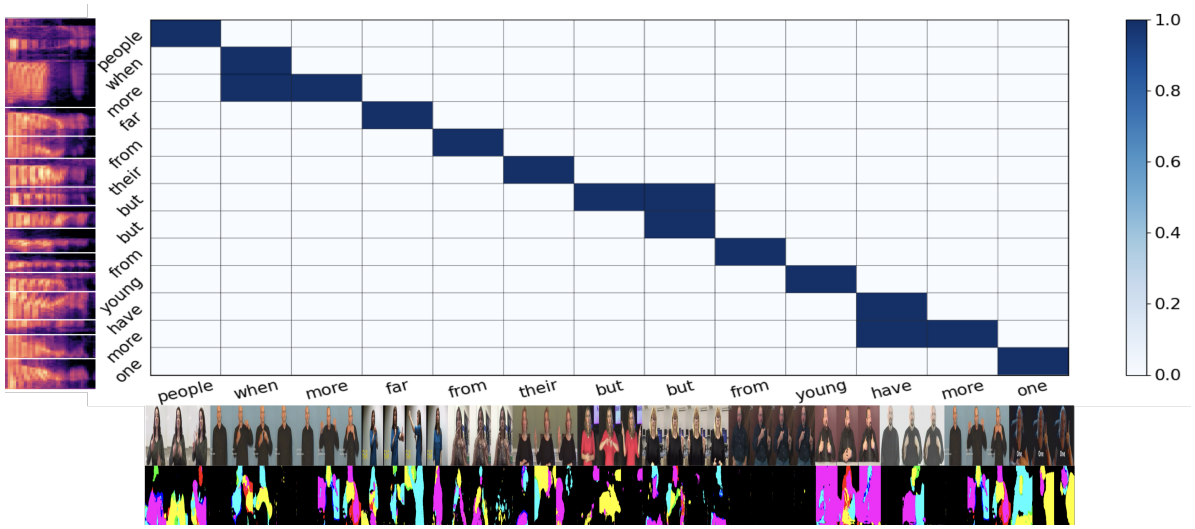


Figure 5: An example of the DTW alignment by speech2sign-U between a pair of speech and sign video (with its optical flow sequence shown below)

land, 1997), 3D-CNN (Huang et al., 2015), two-stream inflated 3D (I3D) CNN (Carreira and Zisserman, 2017; Joze and Koller, 2019), and transformer (Boháček and Hrúz, 2022), among others. To handle multi-sign video sequences, (Koller et al., 2016, 2017, 2018, 2019) reformulate the problem as a sequence labeling problem and develop various systems based on 2D-CNN-HMM hybrid models for German sign language recognition. Later works improve the alignment mechanism of previous models using soft DTW (Huang et al., 2018), CTC with DTW contraints (Pu et al., 2019) or pseudo-labeling refinement (Zhou et al., 2019). While some aim to directly use raw RGB images or generic action features like optical flow as inputs (Koller et al., 2016; Huang et al., 2018; Joze and Koller, 2019), others have found domain-specific features like whole-body and hand keypoints to be more reliable and robust (Boháček and Hrúz, 2022). Thanks to the rapid development of the field, there are now many word-level and sentence-level datasets available in different SLs, and we refer to (Joze and Koller, 2019) for a more comprehensive review.

Unsupervised cross-modal alignment The task of translating between two languages without parallel corpora has been demonstrated between written language pairs (MT-U) and between spoken-written language pairs (ASR-U). (Haghighi et al., 2008) and (Ravi and Knight, 2011; Pourdamghani and Knight, 2017) are respectively the first to treat word-level and sentence-level MT-U as a distribution matching problem and built the first such systems by training statistical machine translation systems using nonparallel corpora, which are further improved by (Artetxe et al., 2018b). To allow more general source and target distributions, (Zhang et al., 2017a,b; Conneau et al., 2018; Artetxe et al., 2018a; Lample et al., 2018) instead use neural networks to embed the source and target distributions and match the distributions using either shared denoising autoencoder (Artetxe et al., 2018a), earth-mover distance minimization (Zhang et al., 2017b) or a generative adversarial network (GAN) with additional regularization losses (Zhang et al., 2017a; Conneau et al., 2018; Lample et al., 2018). (Chung et al., 2018; Liu et al., 2018; Chen et al., 2019; Baevski et al., 2021; Liu et al., 2022) adapt and perfect the GAN-based approach for spoken-written language pairs by leveraging large-scale self-supervised speech representation learning models (Chung and Glass, 2018; Baevski et al., 2020) as well as iterative self-training techniques (Liu et al., 2018).

6 Conclusion

In this paper, we propose the task of unsupervised speech-to-sign language recognition and a neural network model, speech2sign-U, capable of both character-level and word-level SSR-U. On various unpaired speech and ASL datasets, our models consistently outperform previous unsupervised models such as wav2vec-U. Further, we found our model reliable to train for a variety of vocabulary sizes and robust against various types of noise in both speech and visual modalities.

7 Limitations

Our model currently requires high-quality word boundaries for both speech and sign videos. However, as demonstrated by our preliminary results in Table 5, we can overcome such limitations by incorporating more powerful unsupervised segmentation algorithms to our system. Further, while our dataset is sufficient to model the variability in speech and videos, all experiments to date have assumed that spoken and signed sentences share similar word order, which may not be true of natural spoken and signed communications. A future direction of this research will seek to develop methods for spoken-sign language pairs with very different syntactic structures. Lastly, the vocabulary size under our study on word-level SSR-U is relatively small (<1000), and a promising future direction is to extend the current approach to deal with much larger vocabulary size in more diverse conversations.

8 Ethical considerations

One potential ethical concern for our model is the risk of miscommunication. Due to the small amount of resources used to train our system, it tends to be less accurate than its supervised counterpart, and its mistakes may cause confusion, misunderstanding and other psychological harm to the users of our systems. The other ethical concern is that the data used to train the system is demographically homogeneous, as we have noticed from some brief inspections that most of the signers in the ASL datasets are white middle-aged adults. This may lead the system to worse retrieval accuracy for people underrepresented in the training corpus, such as black people, children and elderly people.

Acknowledgement

This work utilizes resources supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729 (Kindratenko et al., 2020), as well as the University of Illinois at Urbana-Champaign. This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics)

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition. In *Neural Information Processing System*.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Neural Information Processing System*.

S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velázquez, and N. Dehak. 2021. Segmental contrastive predictive coding for unsupervised word segmentation. In *Interspeech*, page 366–370.

Matyáš Boháček and Marek Hrúz. 2022. Sign pose-based transformer for word-level sign language recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*.

C. Charayaphan and A. E. Marble. 1992. Image processing system for interpreting motion in american sign language. *Journal of Biomedical Engineering*, 14(5):419–425.

Kuan-Yu Chen, Che-Ping Tsai, Da-Rong Liu, Hung-Yi Lee, and Lin shan Lee. 2019. Completely unsupervised speech recognition by a generative adversarial network harmonized with iteratively refined hidden markov models. In *Interspeech*.

Yu-An Chung and James Glass. 2018. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. In *INTERSPEECH*.

Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass. 2018. Unsupervised cross-modal alignment of speech and text embedding spaces. In *Neural Information Processing System*.

A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Santiago Cuervo, Adrian Lańcucki, Ricard Marxer, Pawel Rychlikowski, and Jan Chorowski. 2022. Variable-rate hierarchical cpc leads to acoustic unit discovery in speech. In *Neural Information Processing System*.

Toni Giorgino. 2009. Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software*, 31(7):1–24.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Neural Information Processing System*.

Gary J. Grimes. 1983. *Digital data entry glove interface device*. US Patent.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 771–779, Columbus, Ohio. Association for Computational Linguistics.

David Harwath, Adrià Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. 2018. Jointly discovering visual objects and spoken words from raw sensory input. In *ECCV*.

Jie Huang, Houqiang Li Wengang Zhou, and Weiping Li. 2015. Sign language recognition using 3d convolutional neural networks. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. 2018. Video-based sign language recognition without temporal segmentation. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Hamid Vaezi Joze and Oscar Koller. 2019. MS-ASL: A large-scale data set and benchmark for understanding american sign language. In *The British Machine Vision Conference (BMVC)*.

V. Kindratenko, D. Mu, Y. Zhan, J. Maloney, S. Hashemi, B. Rabe, K. Xu, R. Campbell, J. Peng, and W. Gropp. 2020. HAL: Computer system for scalable deep learning. In *Practice and Experience in Advanced Research Computing (PEARC '20)*, page 26–30.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Oscar Koller, Necati Cihan Camgoz, Hermann Ney, and Richard Bowden. 2019. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Oscar Koller, Sepehr Zargaran, and Hermann Ney. 2017. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In *The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*, pages 4297–4305.

Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. 2016. Deep sign: Hybrid CNN-HMM for continuous sign language recognition. In *Proc. British Machine Vision Conference (BMVC)*, page 1–12.

Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. 2018. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. *International Journal of Computer Vision (IJCV)*, 126(12):1311–1325.

Felix Kreuk, Joseph Keshet, and Yossi Adi. 2020. Self-supervised contrastive learning for unsupervised phoneme segmentation. In *INTERSPEECH*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Alexander H. Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski. 2022. Towards end-to-end unsupervised speech recognition. In *IEEE Spoken Language Technology Workshop (SLT)*.

Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, and Lin shan Lee. 2018. Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings. In *Interspeech*.

Akash Nagaraj. 2018. ASL alphabet.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. In *ArKiv*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Trevor Gregory, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith

Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Nima Pourdamghani and Kevin Knight. 2017. Deciphering related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2513–2518, Copenhagen, Denmark. Association for Computational Linguistics.

Junfu Pu, Wengang Zhou, and Houqiang Li. 2019. Iterative alignment network for continuous sign language recognition. In *The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 12–21, Portland, Oregon, USA. Association for Computational Linguistics.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing System*.

Thad Starner and Alex Pentland. 1997. Real-time American sign language recognition from video using hidden markov models. *Motion-Based Recognition*, page 227–243.

Shinichi Tamura and Shingo Kawasaki. 1988. Recognition of sign language motion images. *Pattern Recognition*, 21(4):343–353.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017a. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970, Vancouver, Canada. Association for Computational Linguistics.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017b. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945, Copenhagen, Denmark. Association for Computational Linguistics.

Hao Zhou, Wengang Zhou, and Houqiang Li. 2019. Dynamic pseudo label decoding for continuous sign language recognition. In *International Conference on Multimedia and Expo (ICME)*.

A Appendix

A.1 Reproducibility checklist

All experiments are done on four 16GB NVIDIA V100 GPUs and all models are implemented using Pytorch (Paszke et al., 2019) and Fairseq (Ott et al., 2019).

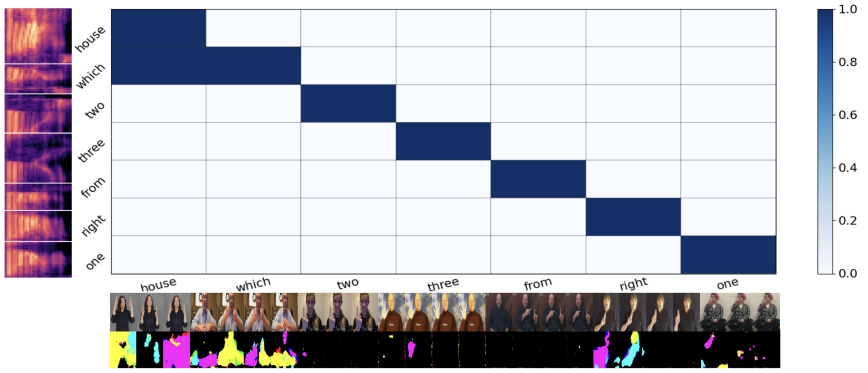
Character-level speech2sign-U We use the exact same generator and discriminator architectures as the wav2vec-U (Baevski et al., 2021). For the CPC-based fingerspelling feature extractor, we use a two-layer MLP as the encoder, with 256 hidden units, ReLU activation and 256 output units and a single-layer LSTM with 256 hidden and output units as the autoregressive predictor. We found 3 prediction steps and 32 negative samples per positive sample for the CPC loss to be the best setting for training. For the CPC-based fingerspelling feature extractor, we train for 60 epochs using Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 0.001, a batch size of 16 with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The checkpoint with the highest average next-frame prediction performance during training is used for the feature extraction later. For the K-means clustering, we use FAISS (Johnson et al., 2019) and set the number of clusters to be the same as the vocabulary size. For the GAN training, we train the model for 10000 updates and validate the model every 1000 updates using the UER metric. We observe similar performance between the best and the last checkpoints for most experiments. Again, we follow the publicly available implementation of wav2vec-U (Baevski et al., 2021) using Fairseq for all the distributed training, optimizer and scheduler setting.

Word-level speech2sign-U For extracting the optical flow features of sign images, we use the OpenCV implementation of Dual TV-L1 method and resized all images to 224×224. For the OpenPose features, we follow the default settings to extract the pose keypoints and set the keypoint coordinates to 0 when the model fails to detect any keypoints. We also normalize the keypoints by the size of the video frame. The I3D model we use are trained on the ImageNet dataset and fine-tuned on the Charades dataset, for both RGB and flow implementations. The same CPC sign encoder as that in character-level experiments is used, except with the pretrained video features as inputs and the outputs of the *MLP encoder* as outputs instead of that of the LSTM model. We then train the CPC sign encoder for 200 epochs on ASL LibriSpeech 1000. The CPC sign encoder features are then quantized into the same number of discrete units as the vocabulary size (100 for ASL LibriSpeech 100, etc.) using K-means implemented in FAISS (Johnson et al., 2019). For the speech feature clustering, we again use the FAISS (Johnson et al., 2019) im-

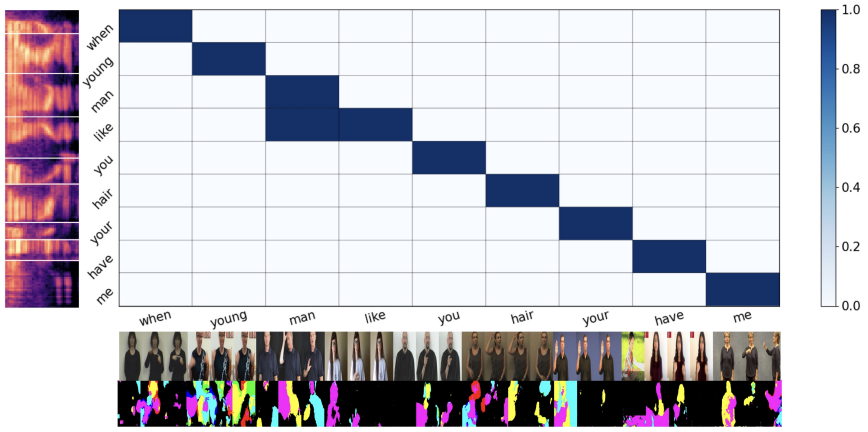
plementation of K-means with a cluster size of about 4 times of the vocabulary size of ASL LibriSpeech 100, 200 and 500, and 2000 clusters for ASL LibriSpeech 1000. The cluster sizes are chosen to ensure a cluster purity of about 90%. For the word-level speech2sign-U, the speech generator is a linear layer with no bias. Skip grams of a maximal step of 6 are used for experiments on ASL LibriSpeech 100, 200 and 500, and a maximal step of 4 are used for ASL LibriSpeech 1000. For the unsupervised training, we train the model for a number of updates equal to $3000 \times \left\lfloor \frac{\text{sample size}}{\text{batch size}} \right\rfloor$. We found that larger batch size generally leads to better performance, and use a batch size of 16k for ASL LibriSpeech 100, 200 and 500, and a batch size of 12k for ASL LibriSpeech 1000 due to GPU memory constraints. Adam optimizer with a initial learning rate of 0.4 and $[\beta_1, \beta_2] = [0.9, 0.999]$ is used throughout the training.

A.2 More SSR-U retrieval examples and error analysis

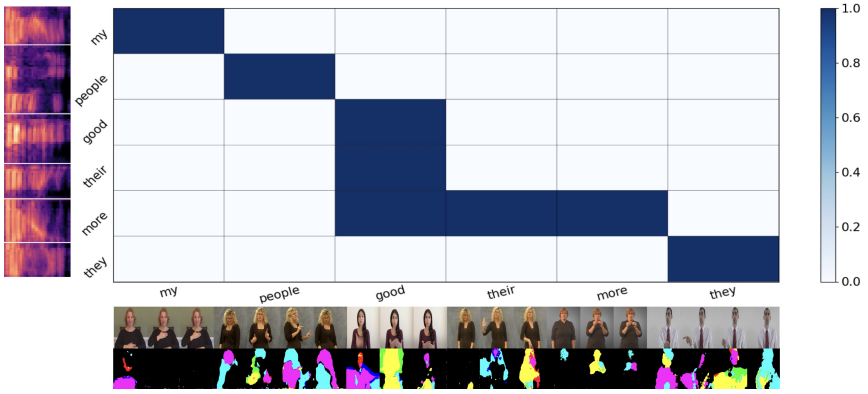
More DTW alignments between speech-video pairs correctly retrieved by speech2sign-U are shown in Figure 6. As we can see, our model is able to correctly align the speech and sign video after the DTW step. However, in order to better understand the type of errors the model is susceptible to, we also show the similarity map *before* the DTW step in Figure 7. While the similarity maps are noisier than their corresponding DTW alignments, the high similarity regions are correctly concentrated approximately along the diagonal most of the time. there are, however, several common failure modes by speech2sign-U. The most common mistake by the model is to confuse less frequent words with more frequent ones, for example, confuse the less frequent word "history" with the more frequent word "from" and "outside" in Figure 7d, or the less frequent "more" with the more frequent "good" in Figure 7c or the less frequent "like" with the more frequent "when" and "man" in Figure 7b. Another type of mistake is to confuse visually similar signs such as "one", "two" and "three" in Figure 7a. The last common type of mistake for speech2sign-U is to confuse acoustically similar words, such as the word "they" and "their" in Figure 7c.



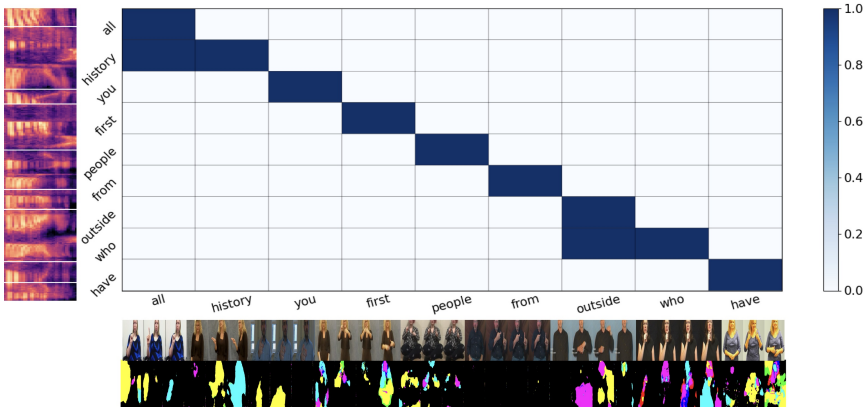
(a)



(b)

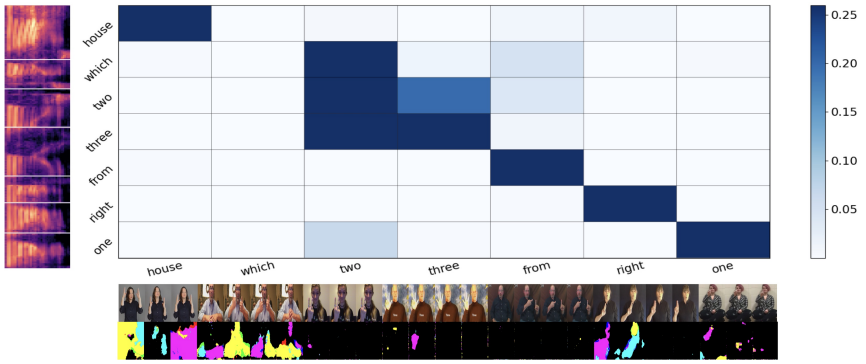


(c)

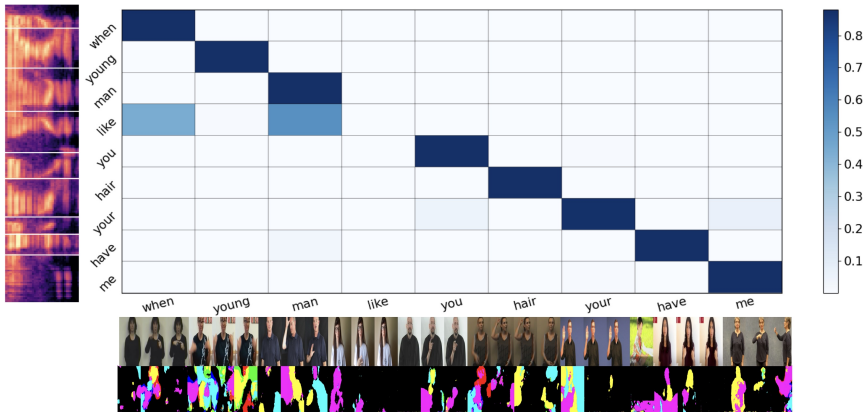


(d)

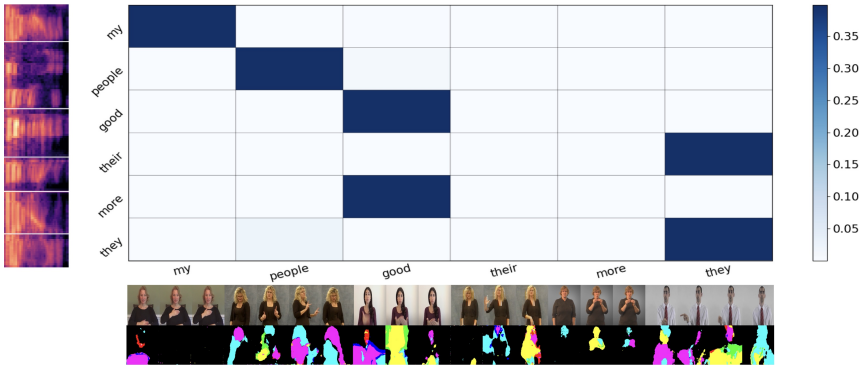
Figure 6: DTW alignments from ASL LibriSpeech 500



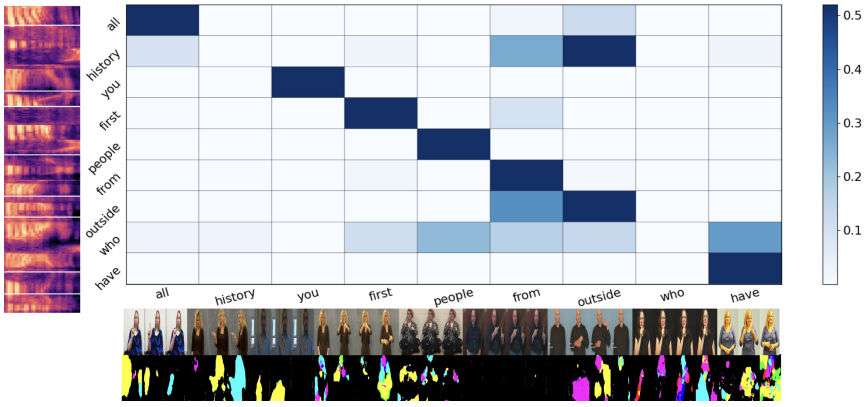
(a)



(b)



(c)



(d)

Figure 7: Similarity maps from ASL LibriSpeech 500

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☑ A2. Did you discuss any potential risks of your work?
Section 8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 2, 3, 4

☑ B1. Did you cite the creators of artifacts you used?
Section 6

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The license and terms of use is straightforward as we use open-source, publicly available software for non-commercial purposes

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. □

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data is collected by other authors and have been carefully checked to remove any privacy-related information

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4, Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4, Appendix A

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix A

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.