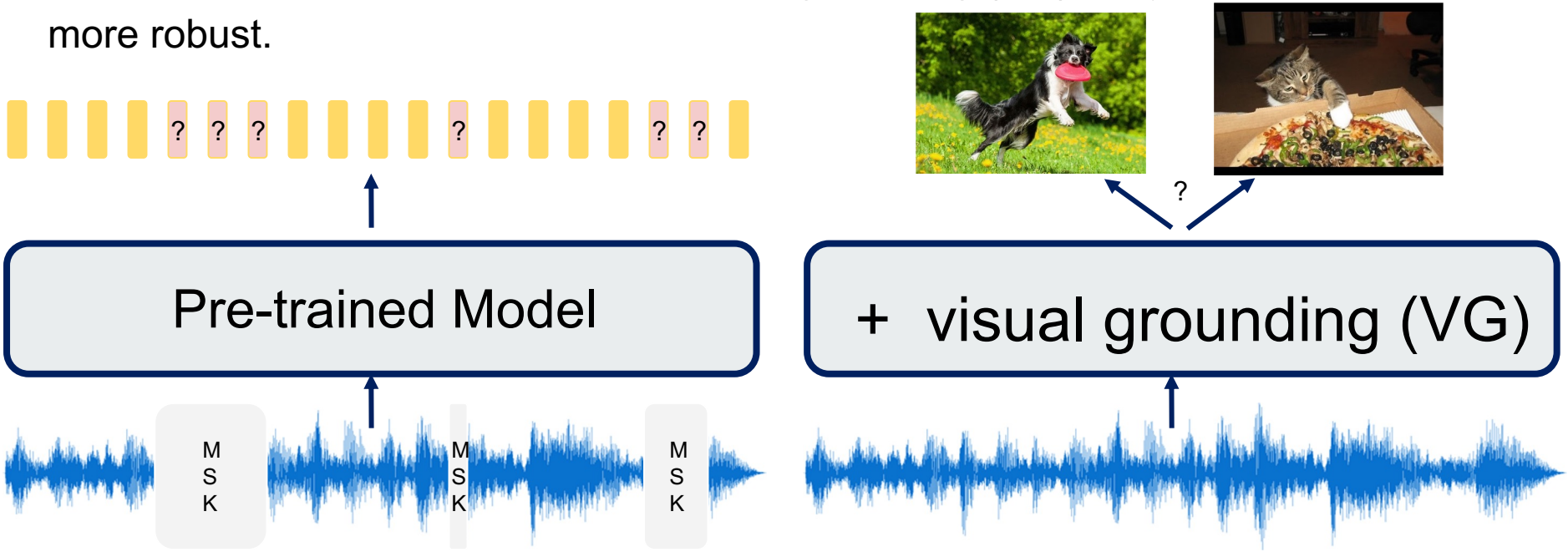
Visually-Enhanced Self-Supervised Speech Models

Team Lead: David Harwath (UT)
Student Team Members: Layne Berry (UT), Heng-Jui Chang (NTU), Po-Heng Chen (NTU), Vanya Cohen (UT), Wei-Yang Lin (NTU), Jason Peng (UT), Ian Shih (NTU), Hsuan-Fu Wang (NTU)

How about adding visual information?

The currently most successful pre-trained models for speech only leverage speech-only information.

However, other modalities, such as visual grounding (VG), may make the models more robust.



Research Directions

SpeechCLIP: Vanya Cohen, Ian Shih, Layne Berry, Hsuan-Fu Wang, Heng-Jui Chang

Word discovery + Star Temporal Classification for Self-Training ASR: Jason Peng

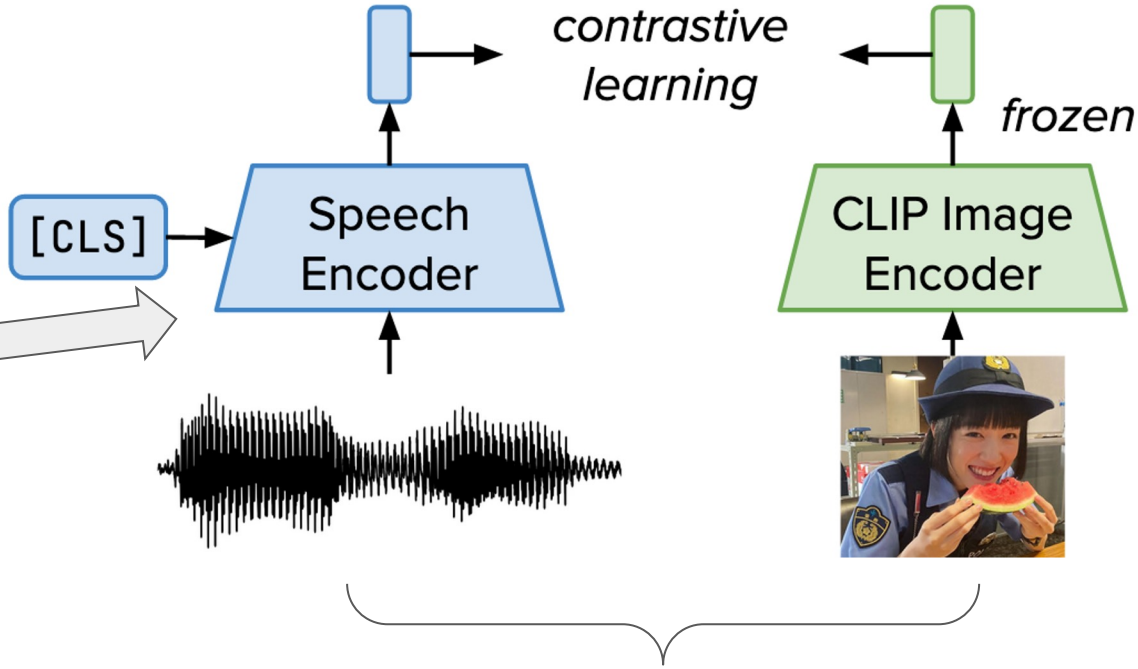
Training w/o negative examples: Po-Heng Chen

Improved grounding/segmentation: Wei-Yang Lin, Heng-Jui Chang

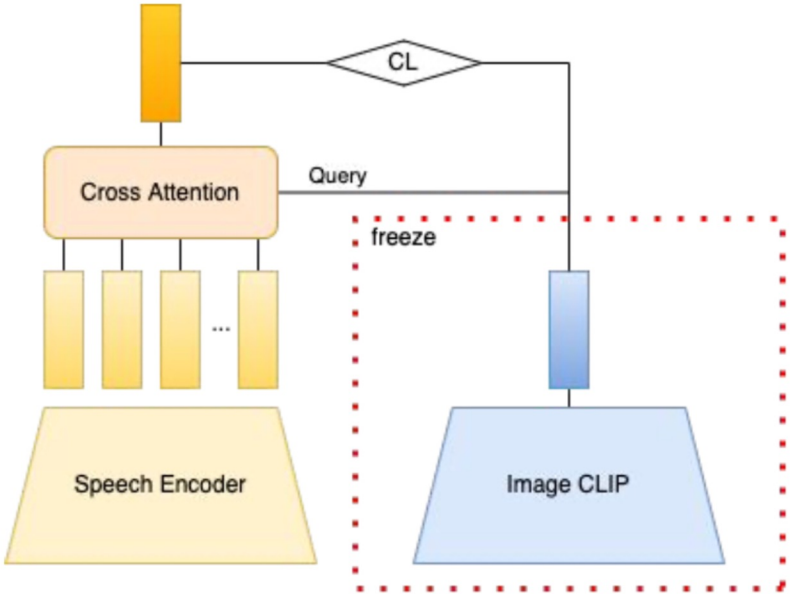
Parallel SpeechCLIP

Use learned representations from this model on downstream tasks (SUPERB, ZeroSpeech)

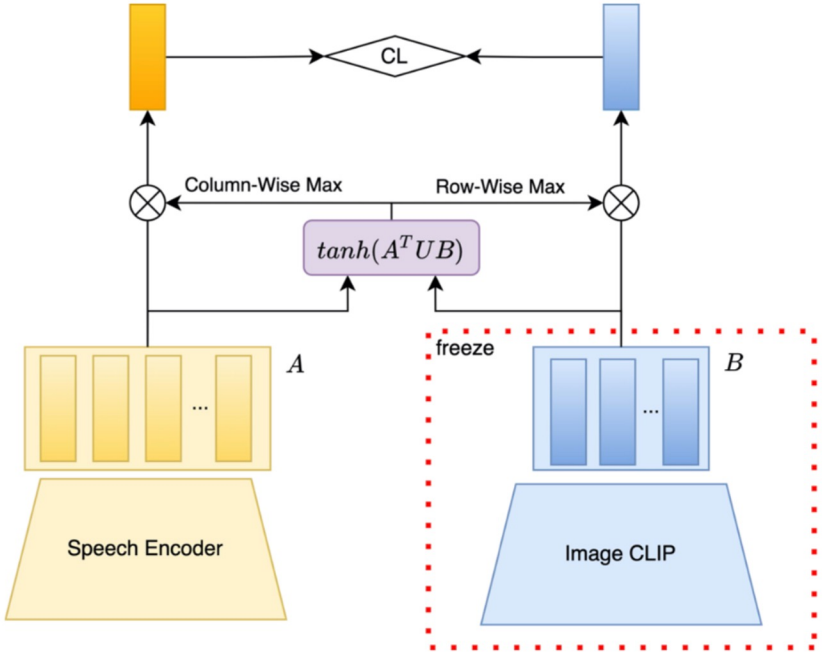
Also evaluate this model's ability to do speech-to-image retrieval vs. models trained from scratch



Training examples from SpokenCOCO, Places Audio, etc…



Cross Modal Attention



Attentative Pooling

Baselines & Experiments - Flickr 8k Test Set

	Audio -> Image			Image -> Audio		
	R@1	R@5	R@10	R@1	R@5	R@10
MFCCs,	8.4	25.7	37.6	12.2	31.9	45.2
MBN Features	12.7	34.9	48.5	16.0	42.8	56.1
MILAN (test)	33.2	62.7	73.9	49.6	79.2	87.5
FAST-VGS (test)	29.3	58.6	71.0	37.9	68.5	79.9
Parallel_Clip + Mean Pool (Wav2CLIP)	14.9	38.9	52.9	20.0	48.1	63.6
Parallel_Clip + [CLS]						
Parallel_Clip + Att Pool	16.7	41.3	55.6	27.8	56.5	70.2
Parallel_Clip + Att Pool (U=I)						
Parallel_Clip + Att Pool Finegrained	5.8	20.1	31.5	5.1	17.8	27.6
CLIP	54.2	80.52	88.42	69.2	91.6	96.1

Zero-shot Audio Caption Flickr 8k (Test)

	Audio -> Text			Text -> Audio		
Models (Caption -> Image)	R@1	R@5	R@10	R@1	R@5	R@10
Parallel_Clip + Mean Pool (Wav2CLIP)	17.7	44.5	60.0	31.8	60.7	76.0
Parallel_Clip + [CLS]						
Parallel_Clip + Attentive Pool	19.9	46.2	60.0	37.8	70.4	80.9

Next steps

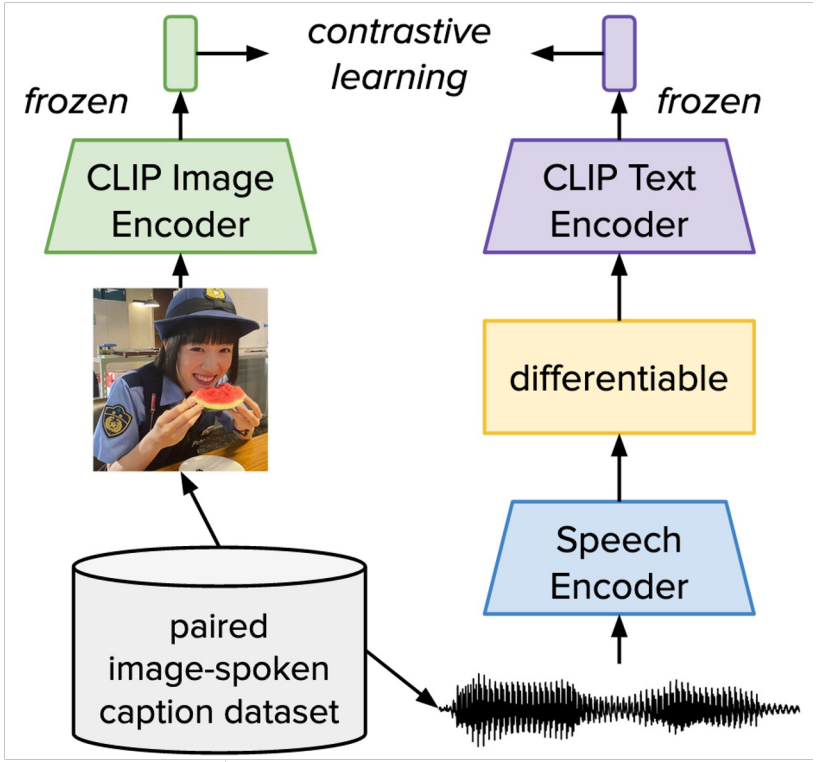
- Interesting results on zero-shot speech-text matching. Can we push this further to estimate partial labels for the speech data?

Cascaded SpeechCLIP

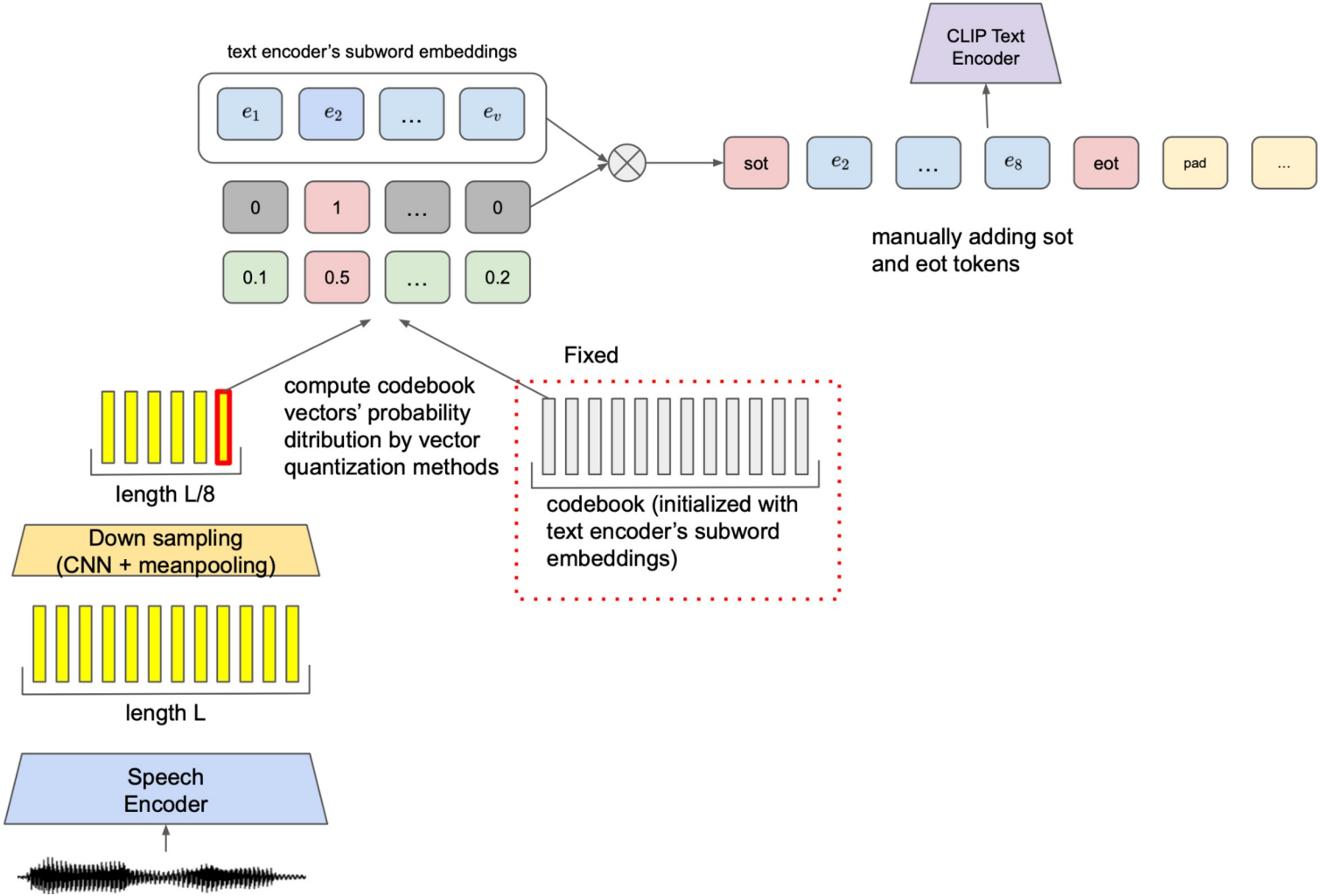
Again, we can use the learned representations from the speech encoder on downstream tasks like SUPERB, ZeroSpeech

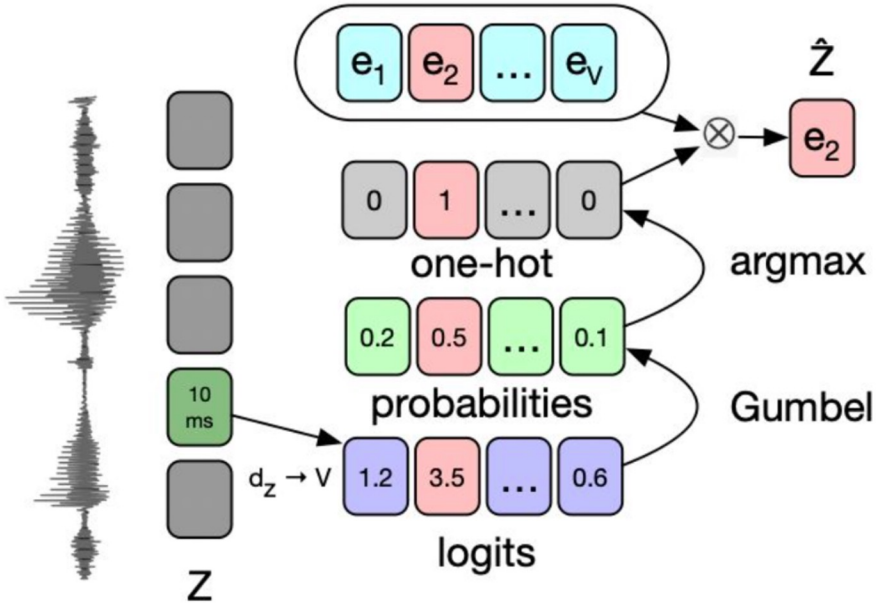
This model also opens the door to some new possibilities:

- Unsupervised speech recognition: Learn to map English speech to English text using the image as a guide
- Unsupervised speech-to-text translation: Use Hindi input speech, learn to map to English text using the image as a guide

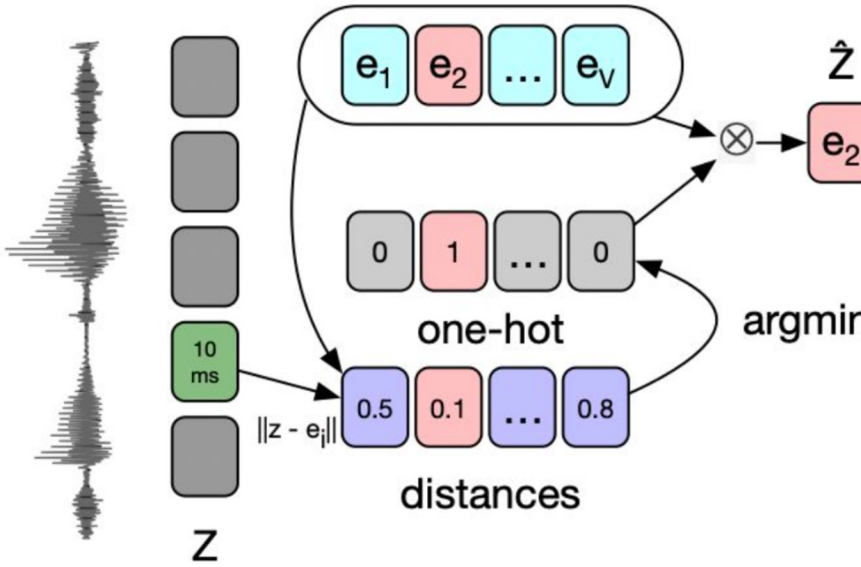


This can be English (SpokenCOCO, Places Audio) or Hindi (Places Audio Hindi)


text encoder's subword embeddings
e_1
e_2
...
e_v
0
1
...
0
0.1
0.5
...
0.2
CLIP Text Encoder
sot
e_2
...
e_8
eot
pad
...
manually adding sot and eot tokens
Fixed
compute codebook vectors' probability ditribution by vector quantization methods
length L/8
codebook (initialized with text encoder's subword embeddings)
Down sampling (CNN + meanpooling)
length L
Speech Encoder



(a) Gumbel-Softmax



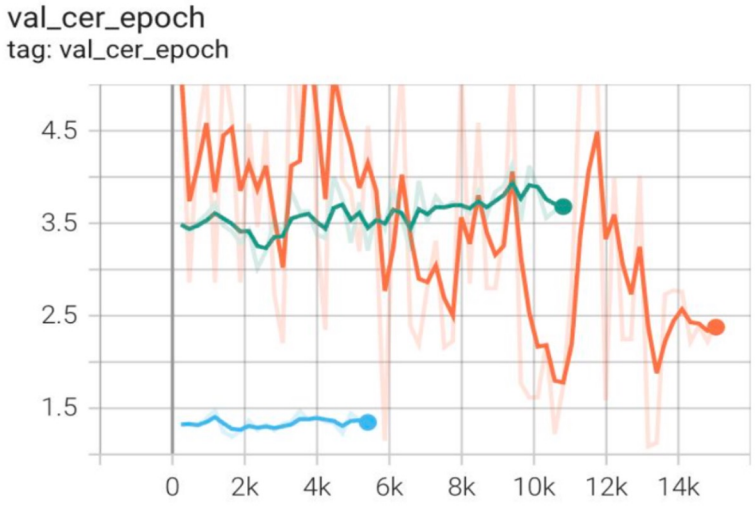
(b) K-means clustering.

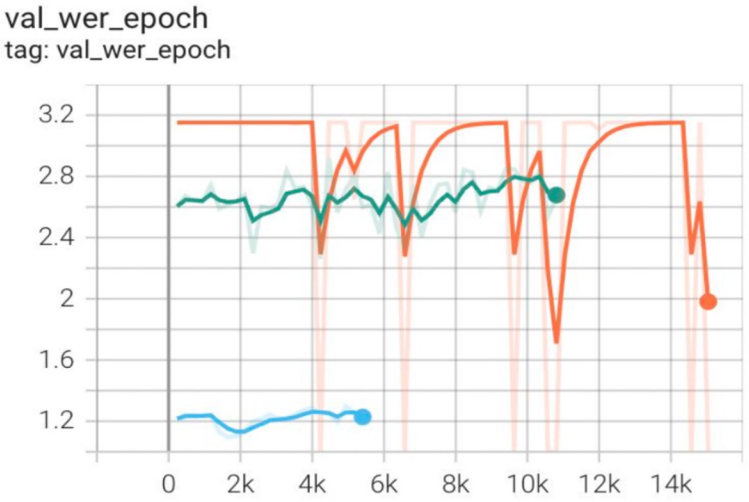
Detokenized text WER & CER

gumbel-downsamplingrate=20

gumbel-downsamplingrate=8

kmeans-downsamplingrate=8





Detokenzied text

gold: "Two small children in red shirts playing on a skateboard"

- kmeans-downsamplingrate=8
  - detok: "tumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumbtumb"
- gumbel-downsamplingrate=8
  - detok: "samsung rapsaver enjoyment shoot ups guns raised anciidentical dusted relatively illa anti midst served glove commercial commercial commercial commercial andi andi andi"
- gumbel-downsamplingrate=20
  - detok: "mercury similar partly solitary client away rays driving rays"

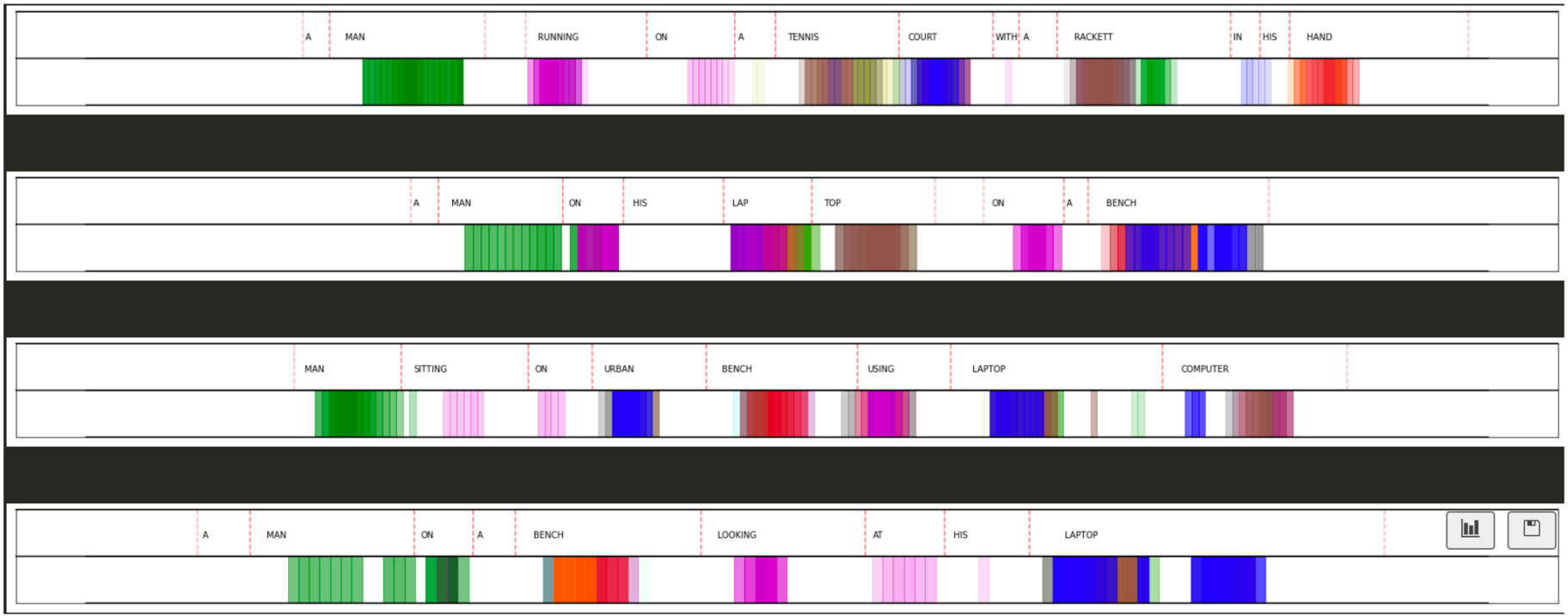
Next steps

- Using non-uniform downsampling (e.g. CIF layers) to produce the input token sequence for the CLIP text encoder

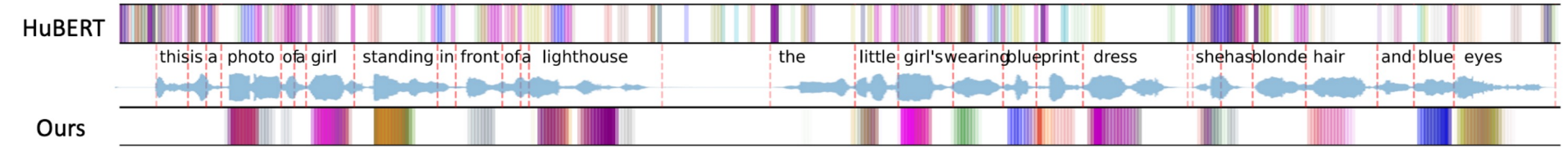
- Bypassing quantization altogether and directly predicting the CLIP input embeddings from the speech encoder

Efficient Self-Training for ASR (Jason Peng)

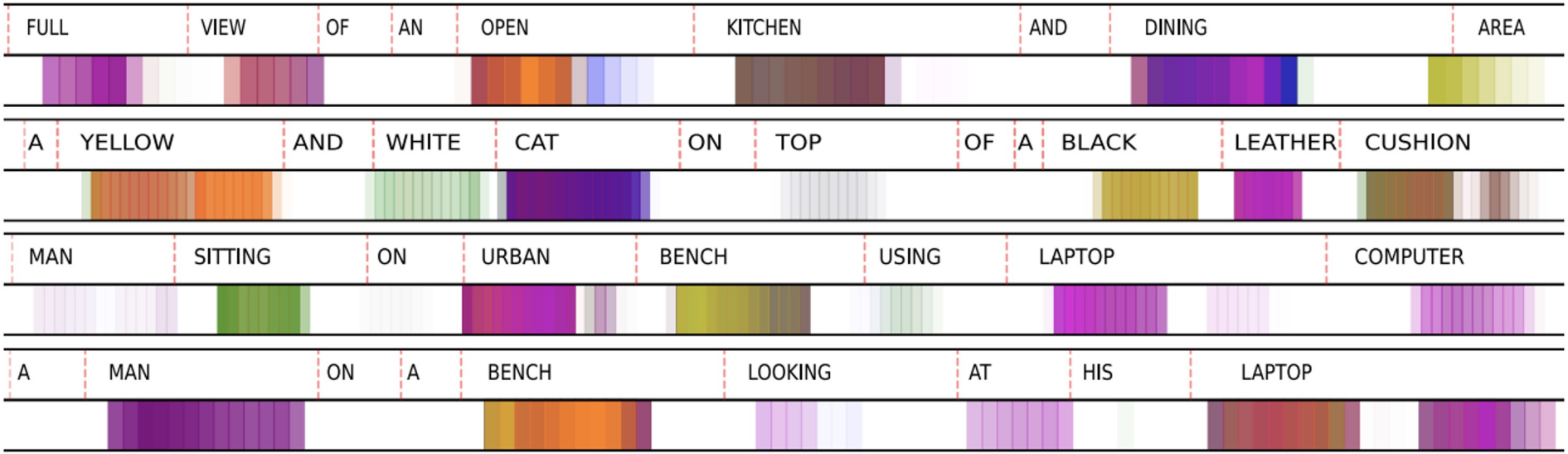
Our latest Transformer-based models for visually grounding speech learn to segment/discover words:



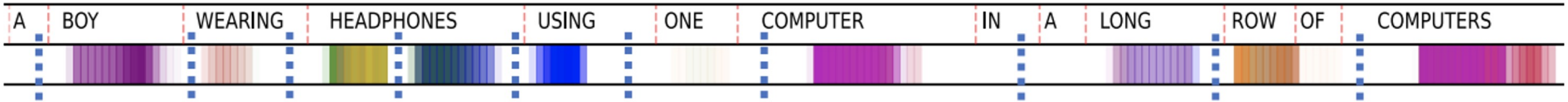
Interesting finding: this same word segmentation ability isn’t present in HuBERT



Detecting and Segmenting Words



Generate word boundaries: midpoints of adjacent boundaries of attention segments:



Word Segmentation on Buckeye and Zerospeech

Table 4: *Results on ZeroSpeech 2020 spoken term discovery.*

	Words	NED	Cov	M-score	Prec.	Rec.	F_1
J.V. [42]	18821	**32.4**	7.9	14.1	32.1	3.2	5.9
Granada et al.	92544	72.3	76.8	40.7	27.8	45.5	34.5
Räsänen et al.	**321603**	52.5	28.7	35.8	25.8	29.9	27.7
ES-KMeans [6]	42473	72.3	**100.0**	43.4	39.6	61.4	48.2
SEA [43]	240033	89.5	99.5	19.0	27.3	75.9	40.1
PDTW [44]	85425	48.2	85.4	64.5	26.5	**88.2**	40.8
VG-HuBERT$_3$ (Ours)	104696	42.5	95.4	**71.8**	**44.7**	54.2	**49.0**

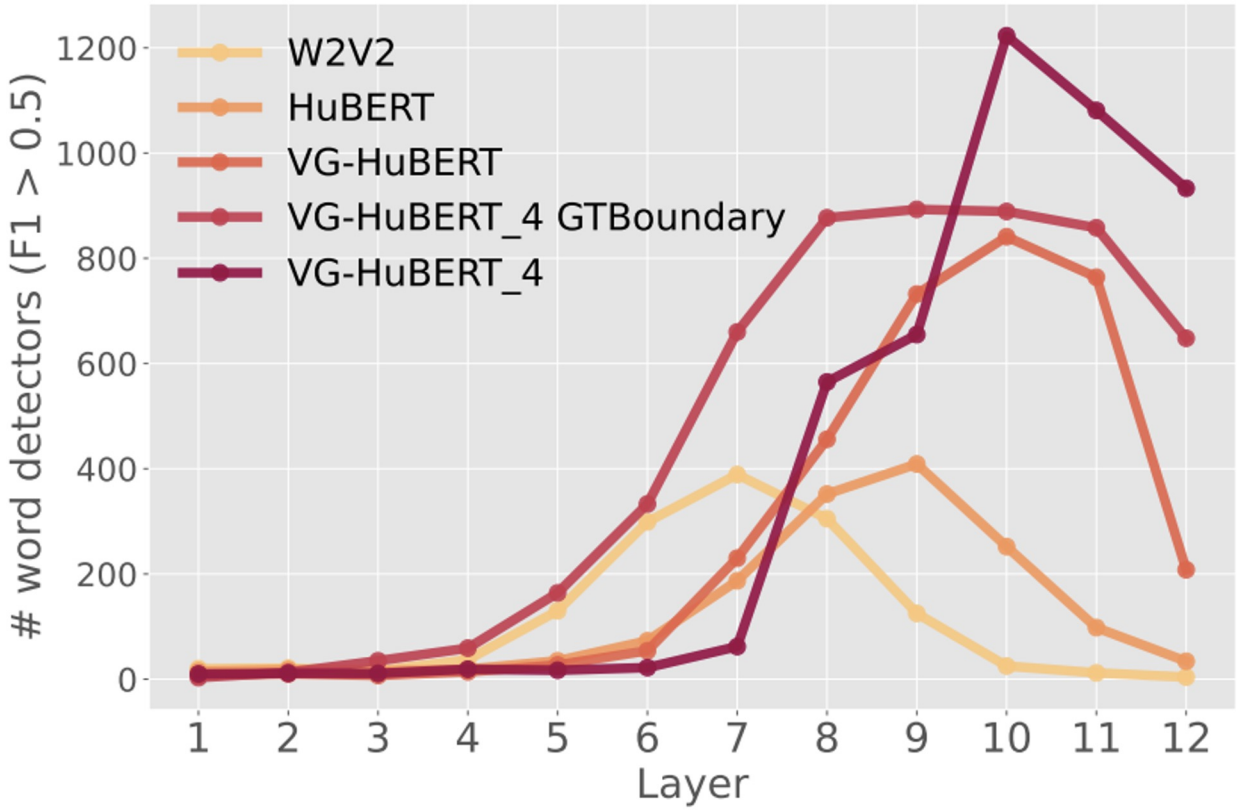
Table 5: *Unsupervised word segmentation on Buckeye test set.*

	Boundary				Token
Model	Prec.	Rec.	F_1	R-val.	F_1
Adaptor gram. [45]	15.9	**57.7**	25.0	-139.9	4.4
SylSeg [46]	27.7	28.9	28.3	37.7	19.3
ES-KMeans [6]	30.3	16.6	21.4	39.1	19.2
BES-GMM [5]	31.5	12.4	17.8	37.2	18.6
SCPC [47]	36.9	29.9	33.0	45.6	-
mACPC [10]	42.1	30.3	35.1	47.4	-
DPDP [8]	35.3	37.7	36.4	44.3	25.0
VG-HuBERT$_3$ (Ours)	**47.6**	42.3	**44.8**	**54.2**	**31.0**

Word Discovery on SpokenCOCO

	Area				Boundary					Word	
Model	WC	tIoU	CD (ms)	A-score	Precision	Recall	F_1	OS	R-val	Purity	WD
ResDAVEnet-VQ [38]	98.39	23.78	119.4	38.30	10.42	50.96	17.30	38.88	-250.77	31.6 ± 0	262 ± 0
W2V2 [39]	77.68	38.56	111.9	51.54	11.52	24.33	15.63	11.12	-33.34	41.2 ± 0.3	407 ± 13
W2V2_{FT} [39]	73.96	39.87	116.1	51.81	11.88	24.79	16.06	10.87	-31.10	43.0 ± 0.1	397 ± 11
HuBERT [40]	72.63	37.72	117.9	49.66	12.18	24.97	16.37	10.51	-28.26	43.1 ± 0.2	404 ± 12
$\text{HuBERT}_{\text{FT}}$ [40]	73.78	39.31	116.0	51.30	11.90	25.81	16.29	11.68	-36.72	45.3 ± 0.3	500 ± 9
FaST-VGS [28]	75.92	54.86	67.8	63.69	28.99	26.17	<u>27.51</u>	-9.72	<u>40.10</u>	70.9 ± 0.5	1011 ± 12
FaST-VGS+ [41]	77.40	51.46	80.5	61.82	22.66	27.86	24.99	22.93	28.54	72.3 ± 0.3	1026 ± 12
VG-W2V2	65.94	45.56	76.8	53.89	18.47	19.78	19.10	7.09	28.86	62.5 ± 0.4	743 ± 14
VG-W2V2_4	71.28	54.06	82.0	61.49	28.15	22.90	25.26	-18.64	39.67	73.0 ± 0.3	<u>1182</u> ± 20
VG-W2V2_5	73.32	53.09	72.2	61.58	28.70	25.45	26.98	-11.32	39.94	75.4 ± 0.1	1149 ± 16
VG-HuBERT	73.16	44.02	89.0	54.97	18.31	18.90	18.60	3.26	29.60	63.2 ± 0.4	823 ± 20
VG-HuBERT_3	70.94	61.29	64.6	**65.77**	35.90	27.03	**30.84**	-24.72	**44.42**	75.3 ± 0.2	1167 ± 26
VG-HuBERT_4	73.97	56.35	78.9	<u>63.97</u>	28.39	25.64	26.94	-9.70	39.64	75.2 ± 0.2	**1230** ± 18

Word Detectors by Layer



Efficient Self-Training for ASR

Rank	Code	Word	F1	Prec	Recall	Occ
2	74	trail	100.00	100.00	100.00	34
3	1045	chewing	100.00	100.00	100.00	18
4	3830	harbor	100.00	100.00	100.00	19
8	211	smart	100.00	100.00	100.00	22
11	3570	shirtless	100.00	100.00	100.00	12
13	1082	garage	100.00	100.00	100.00	19
16	3918	locomotive	100.00	100.00	100.00	20
24	2300	soup	100.00	100.00	100.00	25
31	1367	dirt	98.67	99.33	98.01	148
32	4052	christmas	98.51	97.06	100.00	33
33	2331	boxes	98.36	100.00	96.77	30
34	1931	hydrant	98.17	97.92	98.43	188
35	4077	public	98.00	98.00	98.00	49
36	1269	mouth	97.90	97.22	98.59	70
37	1116	polar	97.73	95.56	100.00	43
38	536	buildings	97.65	99.05	96.30	104
39	505	wine	97.51	96.08	98.99	98
40	2762	batter	97.48	95.08	100.00	58
			...			
180	3181	rural	90.32	93.33	87.50	14
181	1755	jeans	90.32	87.50	93.33	14
182	3492	carrots	90.10	86.67	93.81	91
183	3849	stack	90.00	81.82	100.00	18
186	2522	cakes	90.00	90.00	90.00	18
187	2264	apple	89.92	96.67	84.06	58
188	1089	distance	89.86	83.78	96.88	31
189	2276	team	89.80	84.62	95.65	22
190	1189	colorful	89.74	97.22	83.33	70
191	490	leash	89.66	81.25	100.00	13
192	1372	shopping	89.66	81.25	100.00	13
193	1972	eyes	89.55	90.91	88.24	30
194	1778	kid	89.52	97.92	82.46	47
195	653	tabby	89.47	94.44	85.00	17
196	2378	meal	89.43	82.09	98.21	55
197	934	commuter	89.36	80.77	100.00	21
198	1498	highway	89.36	80.77	100.00	21
199	1859	bottom	89.36	91.30	87.50	21
200	2856	screen	89.31	84.52	94.67	71

We can extract the discovered segments and cluster them (e.g. with K-means) to recover extremely pure word clusters that also have a very high coverage

Current models discover > 1200 different words on SpokenCOCO (out of 6k vocab words) with high accuracy (greater than 0.5 F1)

Big idea: what if you had an untranscribed speech dataset and could cluster words this way, and then have a human annotator listen to just a few examples of each cluster and assign it a label?

You could:

1. Propagate the label to all instances of the cluster across the entire dataset
2. Use these partial transcriptions to train a full ASR system using self-training

Efficient Self-Training for ASR (Jason Peng)

Problem: self-training usually assumes we have a subset of training utterances that are *fully transcribed* and other utterances that are untranscribed. But we only have a bunch of *partially* transcribed utterances.

Solution:

Star Temporal Classification:
Sequence Classification with Partially Labeled Data

Vineel Pratap [1] Awni Hannun [2] Gabriel Synnaeve [1] Ronan Collobert [3]

Original Label: seen from an airplane the island looks like a big spider

Partial Label: from airplane the a spider

Figure 1. An example of speech with a complete and a partial label.

Basic idea: modify CTC training objective to allow wildcards, e.g. training label sequence for above example becomes

```
* from * airplane * the * a * spider *
```

Next Steps

- Finish implementing STC-based ASR training pipeline and perform initial ASR experiments