

A APPENDIX

A.1 BASELINES

AV-cPCFG: We train compound probabilistic context free grammar (cPCFG) (Kim et al., 2019a) on word-level discrete speech tokens. Similar to AV-NSL, word segments are obtained from VG-HuBERT with segment insertion, and segment representations are extracted from VG-HuBERT layer 10 with CLS attention weighted mean-pool. Different from AV-NSL, the segment representations are discretized via kmeans to obtain word-level discrete indices. Because the discretization is word-level instead of phone-level, we swept the number of kmeans cluster over $\{1k, 2k, 4k, 8k, 12k, 16k, 20k\}$, which corresponds to the dictionary size in cPCFG. In summary, AV-cPCFG leverages visual cues only for segmentation and segment representations, but not for phrase structure induction.

DPDP-cPCFG: Instead of training cPCFG on audio-visual word segments and audio-visual segment representations, DPDP-cPCFG does not rely on any visual grounding throughout. Instead, DPDP (Kamper, 2022), a recent speech-only word segmentation algorithm, and vanilla HuBERT representations mean-pooled over DPDP segments are used. We swept through HuBERT layer $\{2, 4, 6, 8, 10, 12\}$. As in AV-cPCFG, kmeans is used for word-level discretization.

oracle AV-NSL: To remove the uncertainty of unsupervised word segmentation, we directly train AV-NSL on top of oracle word segmentation via force alignment. The segment representations are based on learnable attention pooling over vanilla HuBERT layer $\{2, 4, 6, 8, 10, 12\}$ representations. We also tried log Mel spectrograms and HuBERT-L 300M to examine the effectiveness of different input representations. One note is that simpler *score* and *combine* parametrization suffices here⁶.

A.2 HYPERPARAMETERS

For VG-HuBERT, we run MBR selection on the combination of insertion gap $\{0.1, 0.2, 0.3\}$ seconds, segmentation layer $\{9, 10, 11\}$, attention magnitude threshold at top $\{30\%, 20\%, 10\%\}$, three training random seeds, and model snapshots at training step 20k, 30k, 40k, 50k, 60k. This gives 405 combinations in total.

A.3 FULL RESULTS TABLE

A.4 WORD SEGMENTATION VIZ

We show more examples of word segmentation generated by our improved VG-HuBERT in Figure 4. Segments marked with “+” are inserted segments, and vertical blue dotted lines are inferred word boundaries.

A.5 VISUALIZATION OF INDUCED TREES

We visualize the induced trees in Figure 5.

⁶We found that for oracle AV-NSL, the original *score* and *combine* parametrization in VG-NSL works better.

Model			Output	SAIoU
Syntax Induction	Segmentation	Seg. Representation (continuous/discrete)	Selection	
Right-Branching	VG-HuBERT+MBR ₁₀			0.546
Right-Branching	DPDP			0.478
AV-NSL	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₀ (continuous)	MBR	0.516
AV-NSL	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₁ (continuous)	MBR	0.498
AV-NSL	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₂ (continuous)	MBR	0.492
AV-NSL	VG-HuBERT+MBR ₁₀	VG-HuBERT _{10,11,12} (continuous)	MBR	0.521
AV-cPCFG	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₀ +1k km (discrete)	last ckpt.	0.454
AV-cPCFG	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₀ +2k km (discrete)	last ckpt.	0.444
AV-cPCFG	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₀ +4k km (discrete)	last ckpt.	0.499
AV-cPCFG	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₀ +8k km (discrete)	last ckpt.	0.481
AV-cPCFG	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₀ +12k km (discrete)	last ckpt.	0.473
AV-cPCFG	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₀ +16k km (discrete)	last ckpt.	0.471
AV-cPCFG	VG-HuBERT+MBR ₁₀	VG-HuBERT ₁₀ +20k km (discrete)	last ckpt.	0.454
DPDP-cPCFG	DPDP	HuBERT ₂ +1k km (discrete)	last ckpt.	0.434
DPDP-cPCFG	DPDP	HuBERT ₂ +2k km (discrete)	last ckpt.	0.465
DPDP-cPCFG	DPDP	HuBERT ₂ +4k km (discrete)	last ckpt.	0.444
DPDP-cPCFG	DPDP	HuBERT ₂ +8k km (discrete)	last ckpt.	0.387
DPDP-cPCFG	DPDP	HuBERT ₂ +12k km (discrete)	last ckpt.	0.447
DPDP-cPCFG	DPDP	HuBERT ₂ +16k km (discrete)	last ckpt.	0.360
DPDP-cPCFG	DPDP	HuBERT ₁₀ +1k km (discrete)	last ckpt.	0.403
DPDP-cPCFG	DPDP	HuBERT ₁₀ +2k km (discrete)	last ckpt.	0.426
DPDP-cPCFG	DPDP	HuBERT ₁₀ +4k km (discrete)	last ckpt.	0.415
DPDP-cPCFG	DPDP	HuBERT ₁₀ +8k km (discrete)	last ckpt.	0.367
DPDP-cPCFG	DPDP	HuBERT ₁₀ +12k km (discrete)	last ckpt.	0.415
DPDP-cPCFG	DPDP	HuBERT ₁₀ +16k km (discrete)	last ckpt.	0.414

Table 7: Fully-unsupervised phrase structure induction results evaluated with SAIoU.

Model	Segmentation	Seg. Representation	tree target			Output Selection	SAIoU
			train	val	test		
s-Benepar	VG-HuBERT+MBR ₁₀	HuBERT ₂	AV-NSL	AV-NSL	oracle	last ckpt.	0.538
s-Benepar	VG-HuBERT+MBR ₁₀	HuBERT ₄	AV-NSL	AV-NSL	oracle	last ckpt.	0.536
s-Benepar	VG-HuBERT+MBR ₁₀	HuBERT ₆	AV-NSL	AV-NSL	oracle	last ckpt.	0.538
s-Benepar	VG-HuBERT+MBR ₁₀	HuBERT ₈	AV-NSL	AV-NSL	oracle	last ckpt.	0.532
s-Benepar	VG-HuBERT+MBR ₁₀	HuBERT ₁₀	AV-NSL	AV-NSL	oracle	last ckpt.	0.537
s-Benepar	VG-HuBERT+MBR ₁₀	HuBERT ₁₂	AV-NSL	AV-NSL	oracle	last ckpt.	0.536
s-Benepar	VG-HuBERT+MBR ₁₀	HuBERT _{2,4,6,8,10,12}	AV-NSL	AV-NSL	oracle	MBR	0.536

Table 8: Self-training results evaluated with SAIoU.

Syntax Induction	Model		Output Selection	F_1
	Segmentation	Seg. Representation		
Random	oracle			32.77
Left-Branching	oracle			24.56
Right-Branching	oracle			57.39
VG-NSL		word embeddings	Supervised	53.11
AV-NSL	oracle	log-Mel spectrogram	Supervised	42.01
AV-NSL	oracle	HuBERT ₂	Supervised	55.51
AV-NSL	oracle	HuBERT-L ₂₄	Supervised	54.63
AV-NSL	oracle	HuBERT ₂	MBR	54.99
AV-NSL	oracle	HuBERT ₄	MBR	53.25
AV-NSL	oracle	HuBERT ₆	MBR	53.46
AV-NSL	oracle	HuBERT ₈	MBR	53.14
AV-NSL	oracle	HuBERT ₁₀	MBR	36.67
AV-NSL	oracle	HuBERT ₁₂	MBR	48.51
AV-NSL	oracle	HuBERT-L ₂₄	MBR	54.39
AV-NSL	oracle	HuBERT _{2,4,6,8,10,12}	MBR	55.56
AV-NSL	oracle	HuBERT _{2,4,6,8,10,12,24}	MBR	55.96
AV-NSL → s-Benepar	oracle	HuBERT ₂	MBR	57.24
AV-NSL → s-Benepar	oracle	HuBERT ₄	MBR	57.08
AV-NSL → s-Benepar	oracle	HuBERT ₆	MBR	56.81
AV-NSL → s-Benepar	oracle	HuBERT ₈	MBR	56.94
AV-NSL → s-Benepar	oracle	HuBERT ₁₀	MBR	57.16
AV-NSL → s-Benepar	oracle	HuBERT ₁₂	MBR	57.33

Table 9: Phrase structure induction with oracle segmentation given results evaluated with F_1 .

Model	F_1	Constituent Recall			
		NP	VP	PP	ADJP
VG-NSL (Shi et al., 2019)	50.4	79.6	26.2	42.0	22.0
VG-NSL + HI	53.3	74.6	32.5	66.5	21.7
VG-NSL + HI + FastText	54.4	78.8	24.4	65.6	22.0
AV-NSL (oracle seg. + HuBERT ₂)	55.6	55.5	68.1	66.6	22.1
AV-NSL (oracle seg. + HuBERT ₄)	53.7	57.4	56.8	61.3	21.3
AV-NSL (oracle seg. + HuBERT ₆)	53.9	59.4	55.4	59.3	21.2
AV-NSL (oracle seg. + HuBERT ₈)	53.9	56.0	58.0	64.9	22.5
AV-NSL (oracle seg. + HuBERT ₁₀)	50.6	55.8	48.1	57.0	20.5
AV-NSL (oracle seg. + HuBERT ₁₂)	49.0	62.5	34.4	45.0	17.4

Table 10: Recall of specific typed phrases, and overall F_1 score, evaluated on the SpokenCOCO test split. VG-NSL numbers are taken directly from (Shi et al., 2019). AV-NSL here are trained on oracle segmentation with vanilla HuBERT as the layer representations.

Syntax Induction	Model		Visual Embedding	F_1
	Segmentation	Seg. Representation		
AV-NSL	oracle	HuBERT ₂	ResNet101	55.51
AV-NSL	uniform	HuBERT ₂	ResNet101	48.97
AV-NSL	oracle	HuBERT ₁₀	ResNet101	50.50
AV-NSL	uniform	HuBERT ₁₀	ResNet101	36.62
AV-NSL	oracle	HuBERT ₂	DINO	55.71
AV-NSL	oracle	HuBERT ₂	random	31.23

Table 11: Top rows: Impact of segmentation quality for AV-NSL with number of words segments known in advance. Bottom rows: Impact of visual embedding for AV-NSL

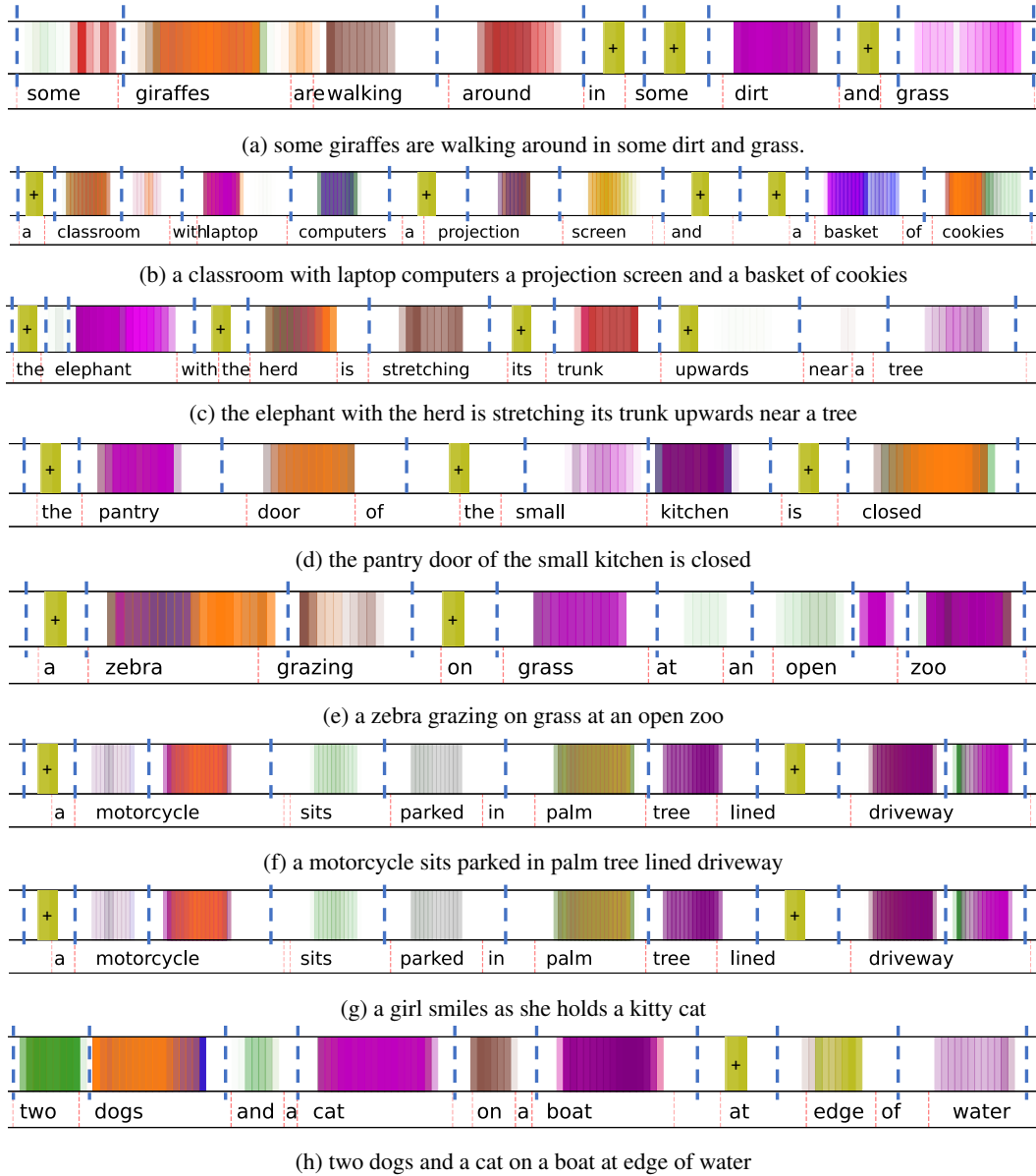


Figure 4: Examples of attention segments generated by VG-HuBERT. Inserted segments are marked with “+”. Vertical blue dotted lines are inferred word boundaries.

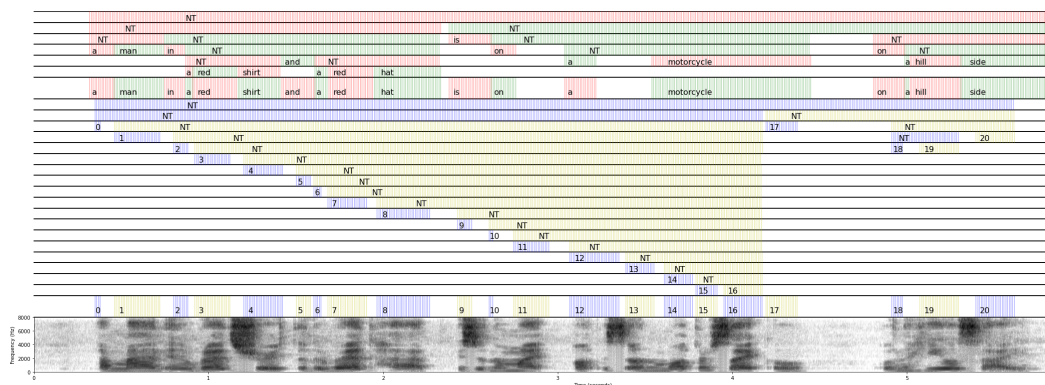


Figure 5: Visualization of an example produced by AV-NSL (best viewed in color). Top (red and green): the ground-truth parse tree; bottom (blue and yellow): the generated parse tree. In each tree, a parent segment adjacently covers its two children segments.