Learning English with Peppa Pig

Mitja Nikolaus
Aix-Marseille University
mitja.nikolaus@univ-amu.fr

Afra Alishahi
Tilburg University
a.alishahi@uvt.nl

Grzegorz Chrupała
Tilburg University
grzegorz@chrupala.me

Abstract

Attempts to computationally simulate the acquisition of spoken language via grounding in perception have a long tradition but have gained momentum in the past few years. Current neural approaches exploit associations between the spoken and visual modality and learn to represent speech and visual data in a joint vector space. A major unresolved issue from the point of ecological validity is the training data, typically consisting of images or videos paired with spoken descriptions of what is depicted. Such a setup guarantees an unrealistically strong correlation between speech and the visual world. In the real world the coupling between the linguistic and the visual is loose, and often contains confounds in the form of correlations with non-semantic aspects of the speech signal. The current study is a first step towards simulating a naturalistic grounding scenario by using a dataset based on the children's cartoon *Peppa Pig*. We train a simple bi-modal architecture on the portion of the data consisting of naturalistic dialog between characters, and evaluate on segments containing descriptive narrations. Despite the weak and confounded signal in this training data our model succeeds at learning aspects of the visual semantics of spoken language.

1 Introduction

Attempts to model or simulate the acquisition of spoken language via grounding in the visual modality date to the beginning of this century (Roy and Pentland, 2002) but have gained momentum recently with the revival of neural networks (e.g. Synnaeve et al., 2014; Harwath and Glass, 2015; Harwath et al., 2016; Chrupała et al., 2017; Alishahi et al., 2017; Harwath et al., 2018; Merkx et al., 2019; Havard et al., 2019; Rouditchenko et al., 2020; Khorrami and Räsänen, 2021; Peng and Harwath, 2021). Current approaches work well enough from an applied point of view, but most are not generalizable to real-life situations that humans or adaptive artificial agents experience.Training data typically consist of images or videos paired with spoken descriptions of the scene depicted. The type of input that a language learner receives from its environment is much more challenging. Firstly, speech is only loosely coupled with the visual modality. Secondly in addition to correlations between the visual scenes and the *meaning* of spoken utterances, there are also correlations with non-semantic aspects of the speech signal, such as the voice of specific speakers, as well as with non-speech ambient sounds. Although it is plausible that such non-semantic correlations can sometimes be useful to the learner in the general endeavour of making sense of the world, for the specific task of learning the semantics of linguistic units they are likely more often an obstacle, as they make it harder to zoom in on the meaning-bearing aspects of the audio signal.

In the current study we make a first step towards simulating the acquisition of language via grounding in perception in a more naturalistic scenario. Our main focus is on learning the meaning of individual words as well as multi-word expressions from spoken utterances grounded in video. We use the well-known children's cartoon *Peppa Pig* as a case study as a source of training and evaluation data. Compared to commonly used video datasets, this data has a number of interesting characteristics. The visual modality is very schematic, and

the language is also simple in terms of vocabulary size and syntactic complexity. Crucially, however, most of the speech in the videos consists of naturalistic dialogs between the characters. The utterances are only loosely and noisily correlated to the scenes and actions depicted in the videos.

GC: Can we support this assertion in a quantitative way? MN: Difficult I think. To measure semantic correlation between images/videos and speech we need to pass them through a model. So we could run our model on a different dataset (e.g. spoken captions) and show that it performs better there, but we wouldn't have directly comparable evaluation metrics..

This choice of data thus allows us to directly address the ecological limitations of the current approaches. In addition, the cartoon videos also contain comments interjected by the narrator. We use these for a evaluation of the acquisition of meaning as they are more descriptive and less noisy and allow us to measure performance while controlling for speaker characteristics. Our contributions are the following:

- We implement a simple bi-modal architecture which learns spoken language embeddings from videos;
- We evaluate model performance in terms of video fragment retrieval and additionally design controlled evaluation protocols inspired by the intermodal preferential looking paradigm (Hirsh-Pasek and Golinkoff, 1996);
- We carry out ablations of model components in order to understand the effects of pretraining for the audio and video encoders, the role of temporal information, and of segmentation strategies while training.

We show that despite the challenges of our naturalistic training data our model succeeds at learning robust associations between spoken forms and their visual semantics. Our findings include the fact that temporal information contributes substantially to video modeling, and that unsupervised pre-training of the audio encoder is key to the best performance, but that even the model trained completely from scratch on about 10 hours of cartoon data performs substantially above chance.

2 Related work

3 Method

The main focus of this study is on the data and evaluation. We thus keep the components of our architecture simple, and follow established modeling practices whenever possible.

3.1 Dataset

We use the dataset provided by Papasarantopoulos and Cohen (2021) containing a set of 209 episodes of the English-language version of *Peppa Pig*. In addition to the raw videos, we also use the annotations created by Papasarantopoulos and Cohen (2021).

These annotations feature written transcriptions aligned with the audio as well as segmentation into *dialog* and *narration*.[1] Dialogs are the parts spoken by the characters, while narrations are comments inserted by the narrator, which are more descriptive in nature. All the narration segments are uttered by the same voice actor. We use the dialogs for training the model, and set aside the narrations for evaluation purposes only. A small portion of the dialog data is also used for validation. Specifically, out of the total 209 episodes, we use dialog from episodes 1–196 for training, and 197–209 for validation. We set aside narrations from episodes 1–104 for validation and 105–209 for testing. Table 1 shows the sizes of the training and validation splits.

Split	Type	Size (h)
train	dialog	10.01
val	dialog	0.66
val	narration	0.94
test	narration	0.64

Table 1: Duration in hours of the dataset splits.

3.2 Preprocessing

Our model is trained to discriminate positive video-audio pairs from negative ones. The positive pairs are those that are temporally coincident in the original video file. In order to generate these training items we need to split the videos into fragments. For segmenting data for training, we *do not* use word or sentence-level subtitle alignment in order to make the setting naturalistic. Processing long segments of video and audio is not tractable on commodity GPU hardware, and we thus segment the data into brief snippets roughly comparable in length to the duration a short sentence or a phrase. We use the following two segmentation strategies:

Fixed Using this approach we simply split sections into fixed-length non-overlapping fragments

[1] It should be noted that the quality of the alignment and segmentation in the original dataset is variable. In cases where exact alignment is needed, such as for word-level analyses, we re-align the transcriptions using `github.com/lowerquality/gentle`.

of 2.3 second duration. This length is close to the mean duration of audio aligned to a single line of subtitles.

Jitter In this approach the mean duration of the segments is the same (2.3 seconds) but we randomly vary the length of the video, and independently, of the corresponding audio around this average duration. This means that (i) the segments can be partially overlapping and (ii) the video and the audio it is paired with are normally of different length. Specifically we sample the fragment duration d (in seconds) from the following distribution:

$$d \sim \min(6, \max(0.05, \mathcal{N}(2.3, 0.5))) \quad (1)$$

The video is subsampled to 10 frames per second, and to 180×100 resolution.[2] The audio is converted to mono by averaging the two channels and the raw waveform is used as input. We use the original sample rate of 44.1 kHz (instead of downsampling to the 16 kHz sample rate used for pre-training WAV2VEC2) as we found out that this helps with generalization performance on the narration validation data.

For evaluation we have a number of different conditions and evaluation metrics described in detail in Section 3.4 and in some of these conditions we use the subtitles to guide segmentation.

3.3 Model Architecture

We adapt the high-level modeling approach from work on spoken image-caption data (Harwath et al., 2016; Chrupała et al., 2017): our objective function is based on a triplet-like contrastive loss with margin which encourages the matching audio and video clip to be projected nearby in the embedding space, and mis-matching audio and video clips to be far away:

$$\ell = \sum_{av} \left[\sum_{a'} \max(0, S_{a'v} - S_{av} + \alpha) + \sum_{v'} \max(0, S_{av'} - S_{av} + \alpha) \right] \quad (2)$$

where α is a margin, S_{av} is a similarity score between a matching audio-video clip pair, and $S_{a'v}$ and $S_{av'}$ denote similarity scores between mismatched pairs, i.e. negative examples from the current batch. Our heuristic to generate positive and negative examples is very simple: we consider the example positive if the audio is aligned with a video clip in our data. Other pairs of audio-video clips are considered negative.

3.3.1 Audio Encoder

The audio encoder portion of the model consists of a `small wav2vec2` model (Baevski et al., 2020) pre-trained in a self-supervised fashion, without any supervised fine tuning.[3] The WAV2VEC 2.0 architecture learns audio embeddings by self-supervised learning driven by a contrastive loss applied to quantized latent representations of masked frames, loosely inspired by the BERT approach to language modeling (Devlin et al., 2019).

The output of this module is a temporal sequence of 28-dimensional vectors. We pool this output across time using an attention mechanism with dimension-wise weights (Merkx et al., 2019):

$$\begin{aligned} \mathbf{A} &= \mathrm{softmax}_t\left(\mathrm{MLP}(\mathbf{X})\right) \\ \mathbf{z} &= \sum_t \left(\mathbf{A}_t \odot \mathbf{X}_t\right), \end{aligned} \quad (3)$$

where $\mathbf{X}$ is the tensor with the encoder output vectors for each time-step t: an MLP followed by a time-wise softmax is used to compute an attention weight for each time step and for each dimension. The pooling is followed by a linear projection and L_2 normalization. For our experiments we also use versions of the encoder where the wav2vec weights are frozen, as well a randomly initialized rather than pre-trained.

3.3.2 Video Encoder

As a video encoder we use the 18-layer ResNet (2+1)D architecture (Tran et al., 2018) pretrained on the action recognition dataset Kinetics-400 (Kay et al., 2017). The pre-trained model is available via Pytorch.[4] This architecture implements 3D convolution by decomposing it into a 2D spatial convolution followed by 1D temporal convolution. The output of this module is aggregated using the attention mechanism with the same architecture as for the audio module, linearly projected

[2] Performance is better with higher resolution (we tried 360×200), but it makes GPU memory requirements prohibitive.

[3] Available from `https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt`.

[4] See `https://pytorch.org/vision/0.8/models.html#resnet-3d`.

to the same dimensionality as the audio (512) and L_2 normalized. For our experiments we also use a version of the video encoder without pre-training.

STATIC baseline As a baseline to investigate the contribution of temporal information to video modeling we swap the video ResNet (2+1)D with the D2 ResNet pre-trained on ImageNet, which embeds each video frame separately. These frame embeddings are then attention-pooled as with the standard video encoder.

3.4 Evaluation

The most common approach to evaluation for visually grounded models trained on spoken image captions is caption-to-image retrieval (often combined with image-to-caption retrieval): in fact this technique has been carried over from text-based image-caption modeling. With the standard spoken caption dataset this approach is unproblematic since the content of the captions is not correlated with extra-linguistic clues in the speech signal, such as speaker identity (since speakers are randomly assigned to captions) or non-speech environmental sounds. In such an artificial setting, a retrieval metric measures the ability of the model to match spoken utterances to images based on their semantic content. This is not the case for the *Peppa Pig* dataset: here we can expect that when a video segment depicts a particular character (e.g. George) then the audio in this segment is more likely to contain utterances spoken by the voice actor playing George. Moreover, some characters might have a tendency to talk about certain topics more often than others, and the model might pick up on these associations instead of paying attention to the actual meaning of the uttered words. Due to these factors, in a naive retrieval setting, a model could obtain a high score by mostly capturing these non-linguistic correlations.

In order to control for these factors we leverage the narrator speech in the videos. These utterances are always spoken by the same actor, so speaker identity cannot be used as a clue for matching video and audio. Furthermore, the narration segments are akin to video captions in that they tend to describe what is happening in the video and thus their semantic content is more strongly correlated with the content of the video than in the case of the dialog, which is also a desirable feature for the purposes of system evaluation.

3.4.1 Video Retrieval

For the retrieval evaluation, as for training, we use the FIXED and JITTER segmentation strategies. We encode each audio clip in a candidate set sampled from the validation (or test) data using the speech encoder part of the model; we encode each video clip using the video encoder. We then measure cosine similarity between the audio clip and all the video clips. If the video clip corresponding to the audio is among the n most similar video clips, we count that as a success. The proportion of successes across all audio clips gives us the retrieval metric known as recall@n: specifically in this paper we focus on $n = 10$. We set the candidate set size to 100, and thus the random baseline for the recall@10 is 10%. In order to quantify uncertainty in this evaluation due to the test data we repeat this procedure 500 times with randomly sampled candidate sets and visualize the score distribution.

3.4.2 Triplets

Retrieval metrics such as recall@10 have some disadvantages. Firstly the absolute value of this metric may be hard to interpret as it depends on the size of the candidate set. Secondly, if we wanted to compare model performance with human performance, we could not feasibly ask human participants to provide the quadratic number of audio-video similarity judgments needed. For these reasons, we evaluate model performance using a more simple and controlled scenario, inspired by intermodal 2-alternative forced choice (2AFC) paradigms in child language acquisition (Hirsh-Pasek and Golinkoff, 1996). The paradigm has been used in language acquisition research to evaluate children's early linguistic knowledge (e.g., Noble et al., 2011; Bergelson and Swingley, 2012), by testing whether they can distinguish a matching (target) visual referent from a foil (distractor) referent when prompted with a word or sentence.

In our case, we extract clips aligned to a single subtitle line, group them by length, and for each pair of same-length video clips[5], we extract the audio from one of them (selected at random) – this is our *anchor*. The video clip from which the anchor was taken is the *positive* one, which the other video clip is the *negative* one. This triplet of stimuli form a single test item. We use the model's

[5]To keep test items independent, the pairing of video clips is done such that each clip only occurs as a member of a single triplet.

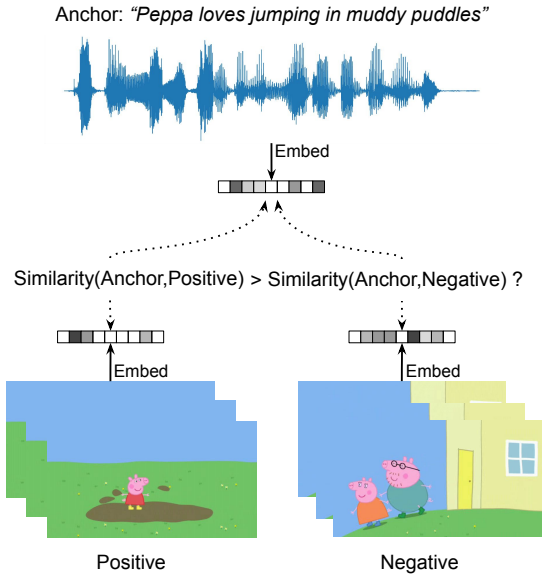

Figure 1: Triplets Evaluation: Given a reference audio sequence (anchor), we measure the model's performance at choosing the matching video (positive) over a random distractor video (negative).

audio encoder to encode the anchor, and the video encoder to encode both video clips. We then check whether anchor is more similar to the positive or negative clips in terms of cosine similarity (see also Figure 1). More precisely, *triplet accuracy* is the mean over all triplets of the following quantity:

$$\frac{\text{signum}(\text{cosine}(A,P) - \text{cosine}(A,N)) + 1}{2} \quad (4)$$

with A being the anchor, P positive and N negative. The triplet accracy metric is inspired by the ABX score of Schatz (2016). For triplet accuracy, regardless of the specific set of test items, we expect random-guessing performance to be at 0.5, and perfect performance to be 1.0. For this metric we also quantify uncertainty by resampling the triplets 500 times from the dataset, and display the score distribution.

3.4.3 Minimal Pairs

While the triplet evaluation gives us a general idea about whether the model has learned a mapping between videos and audio, the metric does not provide insight into whether the model has acquired the grounded semantics of specific words.

This whole section could be shortened.

To address this question, we probe the model's performance in a more targeted setup, where the model is not asked to rank the similarity of a correct video over a random video, but instead over a distractor video with *minimal differences*.

More specifically, this approach considers always pairs of triplets with minimal differences regarding one word in the transcripts of the anchor audios (e.g., *Peppa loves jumping* and *George loves jumping* can be used to test whether the model can discriminate the target word *Peppa* from the distractor word *George*).

We search the transcripts of the validation data for phrases with minimal differences with respect to the most commonly occurring (at least 10 times) nouns, verbs, and adjectives. We set the minimum phrase duration to 0.3 seconds (for shorter sequences, we do not expect that the video data contains enough semantic information for a model to distinguish between target and distractor). A phrase can also be a single word.

Based on each pair of phrases, we create two counter-balanced test trials, an example and a corresponding counter-example, as depicted in Figure 2. Here, the anchor A_{example} of the example triplet is the audio of *Peppa loves jumping*, the positive video P_{example} is the corresponding video, and the negative video N_{example} is the video corresponding to *George loves jumping*. In the counter-example triplet, the anchor $A_{\text{counterex}}$ is the audio of *George loves jumping*, and the positive and negative video are flipped: $P_{\text{counterex}} = N_{\text{example}}; N_{\text{counterex}} = P_{\text{example}}$. By evaluating the model on the examples as well as their corresponding counterexamples, we control the evaluation for linguistic biases in the dataset and ensure that a single-modality model that only considers the audio performs at chance (see also Nikolaus and Fourtassi, 2021).

We measure the model's accuracy for all examples and counter-examples using Equation (4) . Additionally, we report per-word accuracy by calculating the triplet accuracy for all triplets that contain a given word (e.g. *Peppa*) either as target or distractor word, i.e. cases in which the model needs to succeed in either choosing a video containing the given word (the example triplet in Figure 2) or rejecting a video containing the given word (the counter-example triplet in Figure 2). We report accuracy for all words (nouns and verbs) for which we found at least 100 example-counterexample pairs of triplets. There were not enough examples for adjectives in the dataset to perform an evaluation on them.

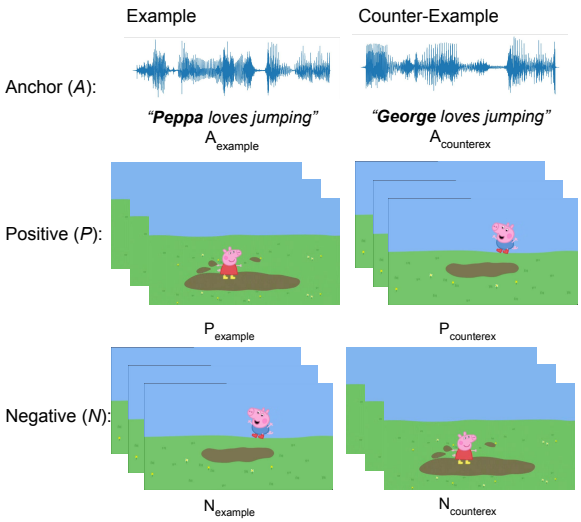

Figure 2: Minimal Pairs Evaluation

4 Experimental Settings

We implement the architecture in PyTorch (Paszke et al., 2019). We use the Adam optimizer (Kingma and Ba, 2015) with the scheduling described in (Devlin et al., 2019). We train every configuration on a single GPU and stop training after 48 hours, with batch-size 8 and accumulating gradients over 8 batches, in 16 bit precision mode. For each model configuration we save model weights after each epoch and report results for the checkpoint which gets the best triplet accuracy on the narration validation data. Our code is publicly available at `github.com/gchrupala/peppa`, and can be consulted for further details of the experimental setup.

Anonymity violation (for the final submission).

5 Results

5.1 Performance metrics

In the following, we present results for the video retrieval and triplet metrics. We report results on both the dialog and narration data. In the case of the narration data the scores are not confounded by speaker-based clues, which is an indication that the model possibly learned to detect some aspects of utterance meaning.

Pre-training and fine-tuning Results on different pre-training configurations are presented in Figure Figure 3. The best overall performance on both the dialog and the narration data is achieved with a model where both the video and audio encoder are pre-trained before being fine-tuned on our data.

A pre-trained video encoder leads to higher gains on the dialog validation data, whereas for model generalization on the narration data we observe that a pre-trained audio encoder is more important.

A model that is trained on scratch using only our data performs still substantially above chance on all metrics (0.1 for recall@10 and 0.5 for triplet accuracy).

To further understand and disentangle the effects of audio pre-training and fine-tuning, we train a model with frozen parameters of the WAV2VEC module (Figure 4). We find that if we don't allow a fine-tuning of the WAV2VEC module, performance decreases substantially on all metrics. In other words, best performance is only achieved with pre-trained and fine-tuned models. For all further experiments we consider this configuration as our baseline for comparison.

MN: Do we maybe want to add the chance (random guessing) baselines into the plots?

Jitter Next, we evaluate a model that has been trained with varying video and audio lengths (JITTER). For fair comparison, we report recall@10 for both FIXED and JITTER validation configurations. As seen in Figure 5, the effect of JITTER is only marginal. Only for the evaluation on dialog data with FIXED clip sizes the performance of the model without JITTER performs slightly better.

Temporal information Finally, we explore the role of the temporal nature of our data, i.e. that our data consists of videos and not static images. Figure 6 compares a model that is trained on the video data with the STATIC baseline. Across all metrics, the we observe substantial performance drops for the STATIC model, which has access to the same data, but does not leverage the temporal information in it.

MN: add significance tests to show that this is the only case with significant differences?

5.2 Minimal Pairs

As a first baseline, we evaluate a model that has been pretrained but not fine-tuned on our dataset. The resulting performance is, as expected, close to chance level: 0.538. Additionally, we evaluate a model that is trained using static (image) data instead of video. The average accuracy is 0.705 . Finally, the best performing model according to the performance metrics (ID 68, audio and video pretraining) achieves an average targeted triplets accuracy of 0.745.

Figure 7 and 8 show per-word accuracy for nouns and verbs, respectively. We perform boos-

GC: The fixed and jitter recall numbers are pretty much the same for these plots. Maybe we should simplify them

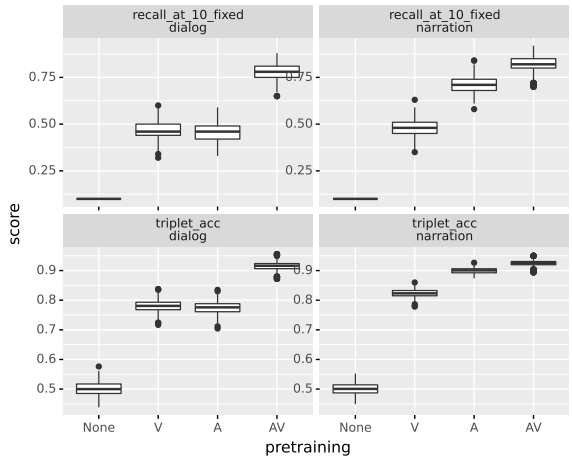

Figure 3: Effect of pre-training.

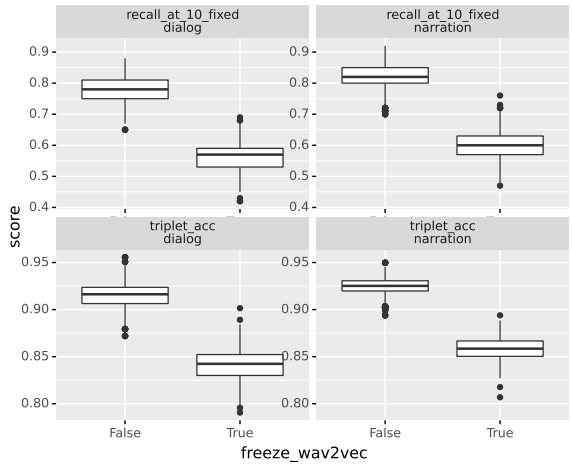

Figure 4: Effect of freezing the parameters of the WAV2VEC module.

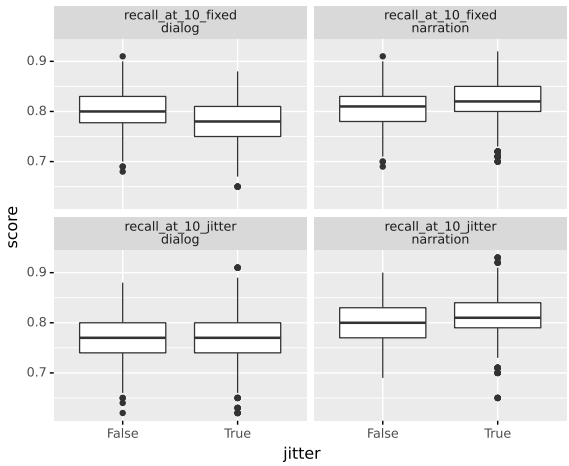

Figure 5: Effect of jitter.

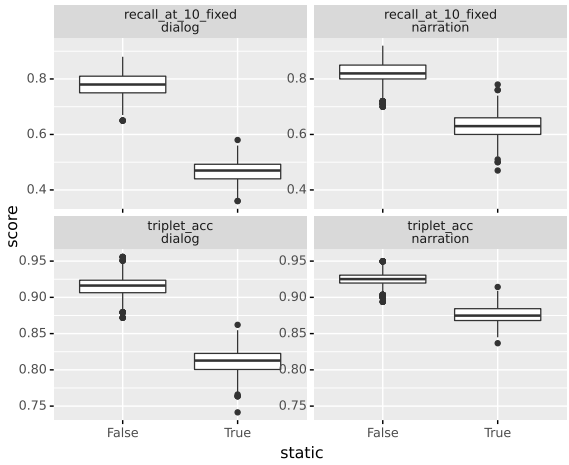

Figure 6: Effect of temporal information.

trapping (n_resampling = 100) to estimate mean and standard deviation for each accuracy score.

We further compute correlations between the per-word accuracy and two possible predictors of age of acquisition: frequency and concreteness. We do not find any significant correlation between the model's per-word accuracy and word concreteness or input frequency of a word in the training data.

MN: verify correlations for final versions

6 Conclusion

Summary of findings.

Potentially discuss:

- Impact of sampling
- Dialogue vs. narration

Future direction:

- Analyses on whether/what the model learns about non-linguistic cues and how it employs them in the retrieval task (e.g. speaker identification, background noise, etc.).
- In-depth (probing-style) analysis on word and sub-word identification.
- Compare static and dynamic models in more detail to see where the dynamic model gains on performance.

References

Afra Alishahi, Marie Barking, and Grzegorz Chrupała. 2017. Encoding of phonology in a recurrent neural model of grounded speech. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL*

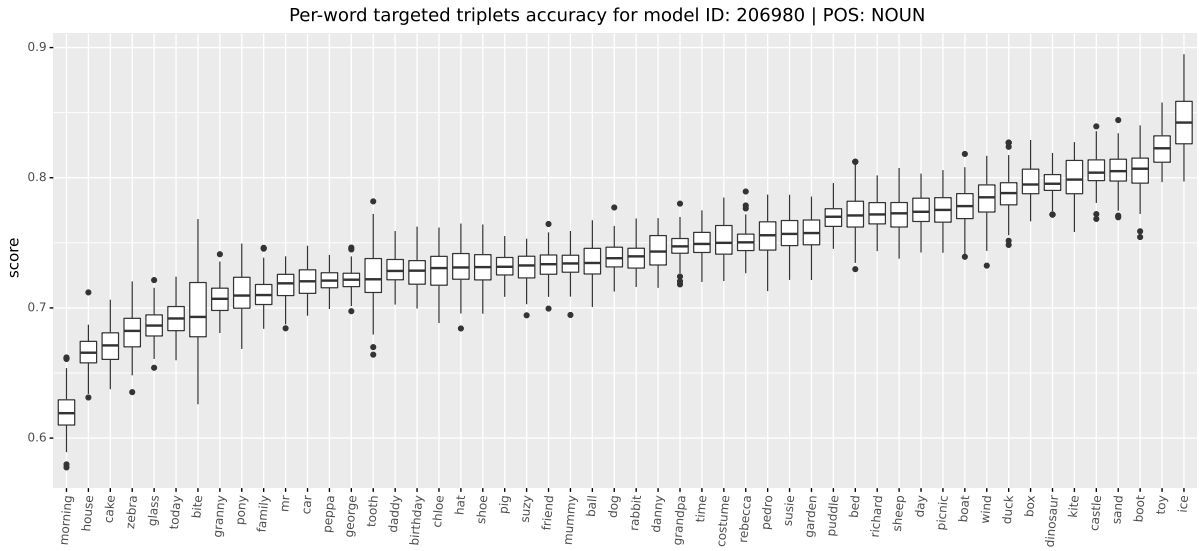

Figure 7: Per-word targeted triplets accuracy for nouns.

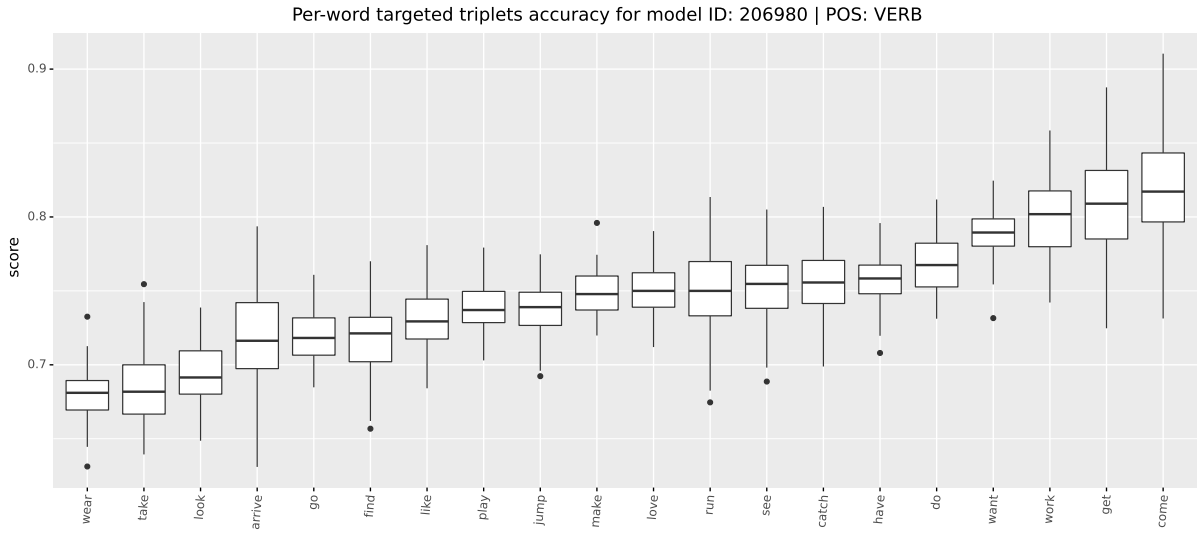

Figure 8: Per-word targeted triplets accuracy for verbs.

2017), pages 368–378, Vancouver, Canada. Association for Computational Linguistics.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Elika Bergelson and Daniel Swingley. 2012. At 6–9 months, human infants know the meanings of many common nouns. *Proceedings of the National Academy of Sciences*, 109(9):3253–3258.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. 2017. Representations of language in a model of visually grounded speech signal. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 613–622, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume*

1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

David Harwath and James Glass. 2015. Deep multimodal semantic embeddings for speech and images. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 237–244. IEEE.

David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. 2018. Jointly discovering visual objects and spoken words from raw sensory input. In *Proceedings of the European conference on computer vision (ECCV)*, pages 649–665.

David F. Harwath, Antonio Torralba, and James R. Glass. 2016. Unsupervised learning of spoken language with visual context. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1858–1866.

William N. Havard, Jean-Pierre Chevrot, and Laurent Besacier. 2019. Models of visually grounded speech signal pay attention to nouns: A bilingual experiment on english and japanese. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 8618–8622. IEEE.

Kathy Hirsh-Pasek and Roberta Michnick Golinkoff. 1996. The intermodal preferential looking paradigm: A window onto emerging language comprehension.

Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The kinetics human action video dataset. *CoRR*, abs/1705.06950.

Khazar Khorrami and Okko Räsänen. 2021. Can phones, syllables, and words emerge as side-products of cross-situational audiovisual learning? - a computational investigation. Preprint `psyarxiv.com/37zn`.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Danny Merkx, Stefan L. Frank, and Mirjam Ernestus. 2019. Language Learning Using Speech to Image Retrieval. In *Proc. Interspeech 2019*, pages 1841–1845.

Mitja Nikolaus and Abdellah Fourtassi. 2021. Evaluating the acquisition of semantic knowledge from cross-situational learning in artificial neural networks. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 200–210, Online. Association for Computational Linguistics.

Claire H Noble, Caroline F Rowland, and Julian M Pine. 2011. Comprehension of argument structure and semantic roles: Evidence from english-learning children and the forced-choice pointing paradigm. *Cognitive science*, 35(5):963–982.

Nikos Papasarantopoulos and Shay B. Cohen. 2021. Narration generation for cartoon videos. Preprint: `https://arxiv.org/abs/2101.06803`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Puyuan Peng and David Harwath. 2021. Fast-slow transformer for visually grounding speech.

Andrew Rouditchenko, Angie Boggust, David Harwath, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, et al. 2020. Avlnet: Learning audio-visual language representations from instructional videos.

Deb K Roy and Alex P Pentland. 2002. Learning words from sights and sounds: A computational model. *Cognitive science*, 26(1):113–146.

Thomas Schatz. 2016. *ABX-discriminability measures and applications*. Ph.D. thesis, Université Paris 6 (UPMC).

Gabriel Synnaeve, Maarten Versteegh, and Emmanuel Dupoux. 2014. Learning words from images and speech. In *NIPS Workshop on Learning Semantics*.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

A Supplementary material

A.1 Retrieval and triplet accuracy

Table 2 and Table 3 show the performance of several model configurations on the retrieval and triplet tasks on the dialog and narration datasets respectively.

GC: Tables out of date

ID	Static	Jitter	Pretraining	Resolution	R@10 (fixed)	R@10 (jitter)	Triplet Acc
68			AV	180x100	0.561	0.533	0.834
206974		Yes	AV	180x100	0.557	0.539	0.830
206964		Yes	AV	360x200	0.594	0.589	0.852
206975		Yes	A	180x100	0.391	0.393	0.744
206976		Yes	V	180x100	0.220	0.205	0.630
206977		Yes	None	180x100	0.194	0.195	0.612
206978	Yes	Yes	AV	180x100	0.442	0.430	0.781

Table 2: Retrieval and triplet scores on dialog validation data.

ID	Static	Jitter	Pretraining	Resolution	R@10 (fixed)	R@10 (jitter)	Triplet Acc
68			AV	180x100	0.644	0.626	0.881
206974		Yes	AV	180x100	0.645	0.623	0.882
206964		Yes	AV	360x200	0.691	0.685	0.904
206975		Yes	A	180x100	0.585	0.574	0.869
206976		Yes	V	180x100	0.316	0.318	0.774
206977		Yes	None	180x100	0.362	0.346	0.770
206978	Yes	Yes	AV	180x100	0.556	0.555	0.850

Table 3: Retrieval and triplet scores on narration validation data.