Learning English with Peppa Pig

Abstract

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 since 2015. As noted by Chrupała (2021), most current approaches work well enough from an applied point of view but leave much to be desired as regards ecological validity. Most datasets consist of static images paired with their descriptions. Existing video datasets contain spoken descriptions of what happens in the video. The type of input that a child faces when learning a language 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 characters and environmental noise. These non-semantic correlations make it harder for the learner to zoom in on those aspects of the audio signal most relevant to learning meanings of linguistic units.

In the current study we make a first step towards simulating such a naturalistic grounding scenario: we use the well-know children's cartoon *Peppa Pig* as a case study. Compared to commonly used video datasets, this data features 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. This choice of data thus allows us to directly address the ecological limitations of the current approaches.

We implement a model which learns to project visual scenes and spoken utterances into a join vector space, and train it on snippets of video containing dialog from the Peppa Pig cartoon, and carry out an in-depth evaluating of the nature of the learned representations using a variety of approaches.

Split	Type	Triplet	Size (h)	Items	Mean length (s)
Training	Dialog	No	9.83	11,058	3.2
Validation	Dialog	No	0.33	375	3.2
Validation	Narration	No	0.80	897	3.2
Validation	Dialog	Yes	0.16	202	2.8
Validation	Narration	Yes	0.45	726	2.2

Table 1: Dataset statistics. For the triplet condition, videos are split such that each segment corresponds to a line of subtitles. For the non-triplet condition, videos are split into 3.2s segments.

2 Related work

3 Method

3.1 Dataset

The dataset consists of the complete set of videos of the English-language version of *Peppa Pig*. In addition to the raw videos we also use the annotation created by (Papasarantopoulos and Cohen, 2021).

These annotations feature written transcriptions of the audio as well as segmentation into *dialog* and *narration*. 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 actor. We use the dialogs for training the model, and set aside the narrations for evaluation purposes only.

Specifically, we use dialog from episodes 1–196 for training, 197–202 for validation and 203-209 for testing. We set aside narrations from episodes 1–104 for validation and 105–209 for testing.

3.2 Preprocessing

For training, we do not use word or sentence level segmentation in order to make the setting more naturalistic. Instead we split the dialog sections into 3.2 second non-overlapping fragments. The video is subsampled to 10 frames per second, and to 180x100 resolution. The audio is converted to mono by averaging the two channels and the raw waveform is used as input.

For evaluation we have a number of different conditions and evaluation metrics described in detail in Section 3.3 and in some of these conditions we use the subtitles to guide segmentation. Table 1 shows the basic statistics of the training and validation splits.

Add the test split.

3.3 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 is has been carried over from text-based image-caption modeling Chrupała (2021). 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 noise. Thus in this setting retrieval measures the ability of the model to match spoken utterances to images based on their semantic content. This not the case for the *Peppa Pig* dataset: here we can expect that when a video segment depics 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. George has a favorite toy dinosaur: when this toy appears in a video segment we can likewise expect higher than random chance of George's voice in the audio. 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 (partially) alleviate these concerns 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 are 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.

Retrieval For the retrieval evaluation we use two conditions: fixed segmentation in 3.2s clips, and segmentation aligned with subtitle line. We encode each audio clip in 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 $n = 10$.

Triplets Retrieval metrics such as recall@10 have some disadvantages. Firstly the absolute value of this metric is hard to interpret as it depends crucially on the size of the candidate set (e.g. the size of the validation/test set). Thus these numbers cannot be directly between different datasets. 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 the following simplified, controlled scenario: We match video clips by length, and for each pair of same-length video clips[1], 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 for a single test item. We use the model's 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. More precisely, *triplet accuracy* is the mean over all triplets of the following quantity:

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

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

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.

3.4 Model

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-loss with margin which encourages the matching audio and video clip to be project 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: namely we consider the example positive if the audio is exactly aligned with a video clip in our data. All other pairs of audio-video clips are considered negative.

The audio encoder portion of the model consists of a small wav2vec model (Baevski et al., 2020) pretrained in an self-supervised fashion, with no fine tuning.[2] During training, we keep the feature extractor and the bottom K transformer layers of this encoder frozen. Its output is pooled across time using a attention mechanism with dimensionwise weights (Merkx et al., 2019):

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

where $\mathbf{X}$ is the tensor with the encoder output vectors for each time-step: an one-hidden layer MLP followed by a time-wise softmax derives a weight for each time step and for each dimension. Each dimension of the pooled embedding vector $\mathbf{z}$ consists of a weighted sum across time of the output values at this dimension. The pooling is followed by a linear projection and L_2 normalization.

As a video encoder we use the 18-layer ResNet 3D architecture (Tran et al., 2018) (not pretrained) as implemented in Pytorch.[3] The output of this module is flattened, linearly projected to the same dimensionality as the audio (512) and L_2 normalized.

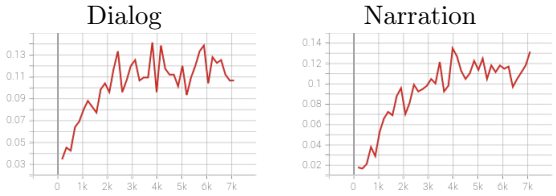

Figure 1: Recall@10 on the retrieval task. The x-axis shows the training steps.

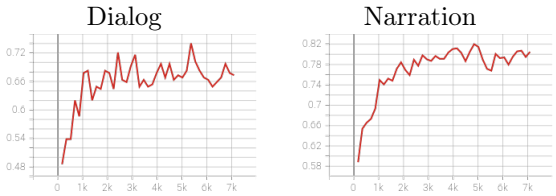

Figure 2: Accuracy on the triplet task. The x-axis shows the training steps.

4 Results

5 Conclusion

References

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Chrupała, G. (2021). Visually grounded models of spoken language: A survey of datasets, architectures and evaluation techniques. Preprint: `https://arxiv.org/abs/2104.13225`.

Chrupała, G., Gelderloos, L., and Alishahi, A. (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.

Harwath, D. F., Torralba, A., and Glass, J. R. (2016). Unsupervised learning of spoken language with visual context. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1858–1866.

Merkx, D., Frank, S. L., and Ernestus, M. (2019). Language Learning Using Speech to Image Retrieval. In *Proc. Interspeech 2019*, pages 1841–1845.

[2]Available from `https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_small.pt`.
[3]Available at `https://pytorch.org/vision/0.8/models.html#resnet-3d`.

Papasarantopoulos, N. and Cohen, S. B. (2021). Narration generation for cartoon videos. Preprint: `https://arxiv.org/abs/2101.06803`.

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

Schatz, T. (2016). *ABX-discriminability measures and applications*. PhD thesis, Université Paris 6 (UPMC).

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (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.