arXiv:submit/3203902 [cs.MM] 1 Jun 2020

Transcription-Enriched Joint Embeddings for Spoken Descriptions of Images and Videos

Benet Oriol, Jordi Luque and Ferran Diego
Telefonica Research
Barcelona, Catalonia / Spain
jordi.luqueserrano@telefonica.com

Xavier Giro-i-Nieto
Universitat Politecnica de Catalunya (UPC)
Barcelona, Catalonia / Spain
xavier.giro@upc.edu

Abstract

In this work, we propose an effective approach for training unique embedding representations by combining three simultaneous modalities: image and spoken and textual narratives. The proposed methodology departs from a baseline system that spawns a embedding space trained with only spoken narratives and image cues. Our experiments on the EPIC-Kitchen and Places Audio Caption datasets show that introducing the human-generated textual transcriptions of the spoken narratives helps to the training procedure yielding to get better embedding representations. The triad speech, image and words allows for a better estimate of the point embedding and show an improving of the performance within tasks like image and speech retrieval, even when text third modality, text, is not present in the task.

1. Introduction

Deep neural networks have become increasingly popular in very different contexts, tasks and modalities, such as vision, audio and language [7]. Multimodal joint embeddings are unified representations for different media types which are generated by modality-specific neural encoders. The parameters of these encodes are often learned with a triplet loss (or similar) that aligns multimodal representations from the same data into the same region of the feature space. For instance, an image of a cat would be mapped by an image encoder to a similar embedding to the one generate by a language encoder of the word ”cat”.

This work explores the gain obtained when adding the textual transcriptions to the joint embeddings learned from egocentric activity images and their spoken narratives. This addition of a the textual modality helps into obtaining richer and more robust representations, which we test in an bi-directional audio from/to video frame retrieval task. This gain is obtained at the cost of collecting the textual transcriptions of the spoken narratives in the training set. Notice though that this transcriptions are not required at test time, as the features are obtained from whether the speech or visual data, only.

2. Previous work

Multimodal features can be of many different kinds. One of the most simple approaches can be concatenating the representations of all modalities into a single multimodal vector. However, this work focuses on *similarity models* that combine different neural encoders that independently map each modality into a single representation space.

DeViSE [11] was one of the first works of similarity models. Their approach consisted of a simple linear transforms that maps from image and text space to a common embedding space, where the inner product or cosine distance between corresponding concepts in different modalities is minimized. Later, Frome et al. [4] proposed more complex mappings to these shared embedding spaces. Pan et al. [8] used a similar approach, but on videos, and using a recurrent neural network. Suris et al. [10] also worked in videos domain with features pooled for each video clip. Aytar et al. [1] proposed a triple modality embedding, using two pair losses, one for the image and text branch and one for the image and audio branch. Their sound branch represent ambient sound, and they use retrieval metrics to prove the improvement of their system with respect to image-sound-text alignement models. Harwath, Torralba et al. [6] used fully convolutional models to map audio descriptions and images into a shared embedding space. Similar to that last one, Harwath, Recasens et al. [5] also mapped speech narrations of images and audio descriptions to a common embedding space, with the main difference that the image and audio feature models do not pool the localized into a single vector, but outputs features that keep the spatial and temporal coordinates.

3. Methodology

This work builds upon the neural architecture proposed by Harwath, Recasens et al. [5]. This architecture is fully convolutional for both the image and speech encoders, which allows localizing the correlations between images and spoken narratives in both space and time . The baseline loss function used by [5] is

$$L = \sum_{i=1}^{B} (\max(0, S_{IA}(I_i, A_j) - S_{IA}(I_i, A_i) + \eta) + \max(0, S_{IA}(I_k, A_i) - S_{IA}(I_i, A_i) + \eta)), \tag{1}$$

being B the size of the minibatch, I_i and A_i the image and audio features of the ith element of the batch, $j, k \neq i$ the impostor indexes, η the margin parameter, and $S_{IA}(I, A)$ a similarity function between image and audio features. This similiarity function is further discussed in Section 3.4.

This training loss is a ranking-based criterion that intuitively aims at maximizing the similarity of corresponding image-audio pairs, and minimizing the similarity of non-corresponding ones. For our approach, and in order to incorporate textual embeddings in the training, we extend the previous equation, that involves image and audio, to three similar instances of the same loss. Each of these three ranking losses corresponds to a modality combination: image and audio, image and text or text and audio. The three loss functions are combined as

$$\begin{aligned} L = \sum_{i=1}^{B} (&\max(0, S_{IA}(I_i, A_j) - S_{IA}(I_i, A_i) + \eta) + \\ &\max(0, S_{IA}(I_k, A_i) - S_{IA}(I_i, A_i) + \eta) \\ &\max(0, S_{IT}(I_i, T_l) - S_{IT}(I_i, T_i) + \eta) \\ &\max(0, S_{IT}(I_m, T_i) - S_{IT}(I_i, T_i) + \eta) \\ &\max(0, S_{TA}(T_i, A_n) - S_{TA}(T_i, A_i) + \eta) \\ &\max(0, S_{TA}(T_o, A_i) - S_{TA}(T_i, A_i) + \eta)), \end{aligned} \tag{2}$$

being $l, m, n, o \neq i$ also impostor indices from inside the minibatch.

This loss function will maximize the similarity between corresponding image-audio pairs, audio-text pairs and image-text pairs and minimize the similiarity between non-corresponding ones. With it, we will be able to train three different branches, mapping corresponding signals from three different modalities into similar points in the embedding space.

3.1. Image Encoder

As proposed in [5], the image encoder follows a VGG-16 [9] architecture, up to the `Conv-5` module, without the maxpool included. On top of that, we add a convolutional layer to map from the 512-dim space VGG output to the desired embedding size space.

3.2. Speech Encoder

As proposed in [5], speech is firstly transformed into log Mel filter bank spectro-grams, with 25ms Hamming window and 10ms time shift. After that, we use a fully convolutional architecture, to map to the embedding space. This model consists of 5 convolutional layers and max poolings.

3.3. Text Encoder

For the text model, we use an off-the-shelf pretrained BERT [3] model, state-of-the-art in natural language feature extraction. We use the Huggingface implementation [1] and the `bert-base-uncased` version.

3.4. Similarity functions

The three models output features that are localized in the space, time or sentence position. In other words, they do not squeeze, respectively, an image, an audio or sentence into single vector. The image model outputs a $N_r \times N_c \times EmbSize$ tensor, being $N_r \times N_c$ the spatial dimensions, and $EmbSize$ the dimensionality of the embedding space. The audio model outputs a $N_a \times EmbSize$ tensor, and the text model a $N_w \times EmbSize$ sized tensor, where N_w and N_a depend on the length of the text and audio. For this reason, computing a similarity between two feature maps is not as straight-forward as, for example, computing a cosine similarity between two vectors.

In order to compute a similarity between them, we first compute a matchmap. The matchmap is the result of computing a cosine similarity between two feature maps in the embedding dimension. More formally, if $I[r, c]$ is the image feature vector at the image embedding width and height r and c, and $A[t]$ is the audio feature vector at audio embedding time t, we can define the image-audio matchmap $M_{IA}[r, c, t]$ with equation 3.

$$M_{IA}[r, c, t] = \langle I[r, c], A[t] \rangle \tag{3}$$

$$M_{IT}[r, c, w] = \langle I[r, c], T[w] \rangle \tag{4}$$

$$M_{TA}[t, w] = \langle T[w], A[t] \rangle \tag{5}$$

being $\langle \cdot, \cdot \rangle$ the inner product operation. Note that while $I[r, c]$ and $A[t]$ are feature vectors (both with size equal to the embedding dimesionality we chose), each element of the matchmap is a scalar value, that expresses the similarity between the image embedding vector at $[r, c]$ and the audio embedding at $[t]$. Similarly to the image-audio matchmap, we will define the text-audio matchmap and image-text matchmap with Equations 5 and 4, respectively.

[1] `https://github.com/huggingface/transformers`

The matchmap can be useful to co-localize concepts in space, time and text, but in order to train the models with our loss, we need to express similarity as a scalar value, getting an overall similarity out of an image-audio matchmap. Harwath, Recasens et al. [5] proposed three different similarity functions for this purpose. This work explore two of them: SIMA (Summation over Image and Maximum over Audio) and MISA (Maximum over Image and Summation over Audio), following Equations 6 and 7:

$$SIMA(M_{IA}) = \frac{1}{N_c N_r} \sum_{r=0}^{N_r - 1} \sum_{r=0}^{N_r - 1} \max_{t} M_{IA}[r, c, t] \quad (6)$$

$$MISA(M_{IA}) = \frac{1}{N_a} \sum_{r=0}^{N_a - 1} \max_{r,c} M_{IA}[r, c, t] \quad (7)$$

Following this idea, we use SIMT and STMA for image-text matchmaps and text-audio matchmaps, represented in equations 8 and 9, respectively.

$$SIMT(M_{IT}) = \frac{1}{N_c N_r} \sum_{r=0}^{N_r - 1} \sum_{r=0}^{N_r - 1} \max_{t} M_{IT}[r, c, w] \quad (8)$$

$$STMA(M_{TA}) = \frac{1}{N_w} \sum_{r=0}^{N_w - 1} \max_{t} M_{TA}[w, t] \quad (9)$$

4. Experiments

The impact of adding the textual transcription of spoken narrations in the learned joint embeddings is assessed for a cross-modal retrieval task in two datasets. The details of the experiments and results obtained are presented in this section.

4.1. Datasets

4.1.1 EPIC Kitchens

The EPIC Kitchens dataset [2] contains egocentric videos of kitchen procedures. Moreover, this dataset also includes the spoken narrations of the actions performed by the user wearing the camera, and clean transcriptions of those narrations in natural language (text).

In order to adapt this dataset for our task, we needed to complete some preprocessing steps. Firstly, as we mentioned, this dataset has actions and speech action narrations. However, they are annotated independently, with no explicit correspondence between both of them. For this reason, we had to align the two sets, using heuristics on the natural language transcription, which is an available field both at the narration annotations and at the action narration annotations. The heuristics included a string comparison (verb stemming, checking string equality and checking inclusion of the narration string into the action string) and a simple alignment algorithm.

With this action-narration correspondence, now we proceeded to segment each kitchen video into smaller clips and the spoken narration into smaller audio clips, using the corresponding timestamps in the action and action narrations annotations. This way, we ended up with tuples of video clips containing a single action and audio narrations containing this single action's description, thus creating corresponding video-audio pairs. Moreover, as we mentioned, each narration also had its clean natural language representation, therefore we actually obtained video-audio-text tuples.

The system works with static image, but now we have video clips. For each video clip we chose N frames, and generate N static image tuples with those. The way we chose those N frames was different in train and validation and test. In the train split, we selected N+2 equally-spaced frames across all video clip, and discarded the first and the last. We did it this way to get the maximum visual diversity among all N frames, but we discarded the first and last since we saw that it normally did not contain much useful information related to the action in the clip. We set N=5 for training. For the validation and test sets, we kept N=1 and select the middle frame in the clip, in order to maximize visual correspondence and keep a small validation set. We will see how validation computation cost grows with $O(2)$ with the validation set size.

With respect to the audio, we decided to adjust the timestamps to 0.3s before the annotated ones, since we observed the beginning of the narrations was being chopped when using the annotated timestamps. Moreover, we limited the length of the narrations to maximum 3s and minimum 0.1s. Also we only took the ones in English.

Before selecting the N frames out of each clip, this left us with a total of 15145 clip-narration pairs. We randomly split the clip-narration pairs into 14.000 training examples, 600 validation examples and 545 test examples. After that, we use the previously explained frame picking rules to obtain 70000 training tuples, 600 validation tuples and 545 test tuples. Again, when we talk about tuples, we refer to image-narration-text tuples.

4.1.2 Places Audio Caption dataset

The Places Audio Caption dataset [6] [5] is a collection of approximately 400k audio caption descriptions obtained via Amazon Mechanical Turk. The described images are from the Places 205 image dataset. This dataset is much larger than EPIC kitchens and has a much bigger variety in concepts and complexity of narrations.

The train split consists of 402385 examples and validation split has 1000 examples.

Text	Dataset	Img R@1	Img R@5	Img R@10	Audio R@1	Audio R@5	Audio R@10
No	EPIC Kitchens	0.178	0.472	0.639	0.182	0.481	0.637
Yes	EPIC Kitchens	**0.193**	**0.514**	**0.686**	**0.183**	**0.494**	**0.677**
No	Places	0.061	0.211	0.312	0.046	0.171	0.262
No	Places *	0.079	0.225	0.314	0.057	0.191	0.291
Yes	Places	**0.116**	**0.278**	**0.394**	**0.072**	**0.24**	**0.338**

Table 1. In this table we present the recall scores in both datasets and with or without adding text at train time. The difference between the 3rd and 4th line (marked with *) is that the 4th are the results published at [5] while the 3th are our results of the same experiment. We used the implementation they provided, but couldn't have the same experimental setup, reducing the batch size from 128 to 80. Results are on test dataset for EPIC and validation for Places

4.2. Training Details

All models were trained with SGD, with learning rate of 0.001, decaying by a ratio of 10 every 70 epochs, and a momentum of 0.9. When not using text, models were trained with the loss function from equation 1 and when using text, models are trained with the loss function from equation 2. Regarding the similarity functions, we used SIMA, SIMT and STMA for EPIC kitchens and MISA, MIST and STMA for the Places dataset. For EPIC kitchens, we used a batch size of 30 and for Places a batch size of 80. This because we used machines with more computational resources to run the experiments of the biggest dataset. The BERT embedding was kept frozen, so the weights were not updated. The size of the embedding was set to 768, to match the BERT output dimensions and did not need to train anything on top of it.

4.3. Cross-modal Retrieval

The task used to assess the quality of the embeddings was image and audio retrieval. In other words, we computed the similarity between all audios and all images in the validation split, and perform speech narrations retrieval queried by image and retrieval of image queried by speech narrations, using the similarity as ranking score to retrieve images.

Table 1 shows that adding text at training time improves all recall scores. We think this is due to two main reasons. Firstly, textual transcriptions can be understood as a clean version on audio data, and it encodes in a much cleaner way the concept in the video clip. For this reason, it is no surprise that it helps as an intermediate modality between audio and image. Secondly, pretrained BERT is the state-of-the-art in text embeddings. This model had already discovered concepts during its pretraining. For this reason, we do not need the audio and image branches (trained from scratch) to *discover* the concepts on their own, and find BERT as almost a ground truth embedding they need to learn the mapping to.

5. Conclusions

This work shows how textual transcriptions can enrich the quality of the representations learned for joint embedding spaces between images and speech. The textual representations improve the image and speech encoders, as they help to train more robust image and speech embeddings that improve the recall metrics in the tasks of speech narration retrieval queried by image, and viceversa. We also point out that the egocentric nature of speech and vision, could help the retrieval task.

Acknowledgements

This work was supported by the the Industrial Doctorate 2017-DI-011 from the Government of Catalonia, and the Spanish Ministry of Economy and Competitiveness through the European Regional Development Fund (TEC2016-75976-R). We gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs used in this work.

References

[1] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. See, hear, and read: Deep aligned representations. *arXiv preprint arXiv:1706.00932*, 2017.

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

[4] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances*

in neural information processing systems, pages 2121–2129, 2013.

[5] David Harwath, Adrià Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James R. Glass. Jointly discovering visual objects and spoken words from raw sensory input. *CoRR*, abs/1804.01452, 2018.

[6] David Harwath, Antonio Torralba, and James Glass. Unsupervised learning of spoken language with visual context. In *Advances in Neural Information Processing Systems*, pages 1858–1866, 2016.

[7] Xavier Giro i Nieto. One perceptron to rule them all: Language, vision, audio and speech (tutorial). In *Proceedings of the 2020 on International Conference on Multimedia Retrieval*, 2020.

[8] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[10] Didac Surís, Amanda Duarte, Amaia Salvador, Jordi Torres, and Xavier Giró-i Nieto. Cross-modal embeddings for video and audio retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[11] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. In *European Conference on Machine Learning*, 2010.