arXiv:2011.09634v1 [cs.CV] 19 Nov 2020

Watch and Learn: Mapping Language and Noisy Real-world Videos with Self-supervision

Yujie Zhong *
Malong LLC
jaszhong@malongtech.com

Linhai Xie
University of Oxford
linhai.xie@cs.ox.ac.uk

Sen Wang
Heriot-Watt University
s.wang@hw.ac.uk

Lucia Specia
Imperial College London
l.specia@ic.ac.uk

Yishu Miao
Imperial College London
ym713@ic.ac.uk

Abstract

In this paper, we teach machines to understand visuals and natural language by learning the mapping between sentences and noisy video snippets without explicit annotations. Firstly, we define a self-supervised learning framework that captures the cross-modal information. A novel adversarial learning module is then introduced to explicitly handle the noises in the natural videos, where the subtitle sentences are not guaranteed to be strongly corresponded to the video snippets. For training and evaluation, we contribute a new dataset 'ApartmenTour' that contains a large number of online videos and subtitles. We carry out experiments on the bidirectional retrieval tasks between sentences and videos, and the results demonstrate that our proposed model achieves the state-of-the-art performance on both retrieval tasks and exceeds several strong baselines.

1 Introduction

Learning the mapping between vision and language in a supervised manner has been actively studied for many years [12, 13, 28, 15, 29, 18, 3, 24, 10, 31, 35, 30, 25, 33, 34, 4]. However, annotating parallel data costs a large amount of human labor (e.g. [16, 11]). Therefore, we propose to straightforwardly teach machines to understand online videos with transcribed subtitles from speech. Certainly, the subtitles are not the perfect description, but they are naturally aligned with the videos, and a lot of them are explicitly or implicitly grounded by the objects and scenes from the videos. Following the intuition of how humans learn to pick up new knowledge from the real-world, the machine learns to selectively choose the pairs and learn the mapping of the visual and language. In this case, we have access to massive unlabelled video datasets online for visual and language grounding. We collect a large video dataset without annotations, termed the 'ApartmenTour' dataset. These videos are decent source materials for capturing the language-visual correspondence (LVC), since most of them are describing the indoor scenes, objects or surroundings (see Figure 1).

However, learning the LVC on these noisy videos is very challenging, because there is still a lot of subtitles and video frames that are loosely correlated, unlike the parallel dataset of image captioning (e.g. MS-COCO) where the captions are produced by humans to describe the image on purpose. For instance, the author may talk about things irrelevant to the scenes (e.g. the bottom-right example in Figure 1). In addition, the videos provide sequential visual signals instead of a single image with caption, and the sentences have to be mapped over consecutive frames to locate the corresponding ones in a certain video clip, which further complicates the cross-modal learning in this case. Therefore, simply applying encoder-decoder models is not good enough to tackle this problem.

In this work, we propose a framework for learning the LVC by minimising a cross-entropy loss that brings together the subtitle and video pairs. An attention mechanism is employed to dynamically focus on the frames of a video clip that correspond to the subtitle sentence and disregard the irrelevant ones. In addition, we apply an adversarial loss during training such that the network is able to selectively start with learning from simple examples (i.e. video-sentence pairs with obvious correspondence),

*Work done during PhD at Oxford

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

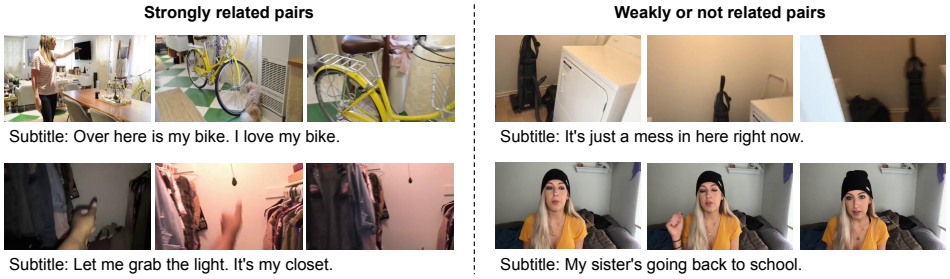

Figure 1: Examples of the 'ApartmenTour' dataset. Three selected frames and the subtitle are shown for each video clip. The top two rows are the strongly-related pairs; whereas the bottom ones can be seen as noises since they are weekly or not related pairs, which makes the learning very challenging.

and then gradually move on to the difficult ones. The experiments quantitatively and qualitatively demonstrate the superiority of the proposed model compared to the strong baselines, and show the great potential of watch-and-learn strategy.

Related work. A related work to this paper is [20] which aims at learning the LVC from a large corpus of narrative videos [21] with self-supervision. While [20] focuses on designing a novel loss function, we propose to dynamically choose training samples based on the learning difficulty.

2 Methodology

2.1 Problem Definition

Language-visual correspondence. The essence is to leverage the fact that a lot of visual scenes, objects and even actions are naturally aligned with the speech in the online videos. The LVC task on the collected 'ApartmenTour' dataset (more details of the dataset is described in Section 3) can be very challenging due to several reasons. First, there is a significant amount of time in each video when the author is not describing the apartment. Second, there also exists a lot of cases where the visuals only contain a small subset of the things described in the speech or only a small portion of the frames corresponds to what the subtitle refers to. The existence of loose-correspondence can severely confuse the system in the training process as they introduce some noises.

Loose-correspondence alleviation. Intuitively, it is easier for the network to learn the language-visual correspondence from the training pairs with relatively obvious correspondence at the beginning of the training, and leave out the less obvious ones which can be involved to the training at later stages. This training scheme shares a similar spirit with the curriculum learning [2] which presents training examples in a meaningful order, i.e. gradually illustrating more complex ones. It is shown to improve the performance of networks in many machine learning tasks, especially in weakly supervised learning [7, 8]. Following this intuition, we propose a network named Watch-And-Learn (*WAL*) to alleviate this loose-correspondence.

2.2 Network Architecture

Vision channel and language channel. The vision channel in the *WAL* network extracts and embeds the visual representation of videos with the frame-level attention. We experiment with three different attention mechanisms in the attention module, namely dot-product [17], multiplicative [17] and additive [1]. The language channel consists of a 12-layer bidirectional transformer [6] and an embedding module. The embedding module embeds the BERT representation to a common space with the embedded visual features, giving s in Figure 2.

Discriminator. The aim of this discriminator (bottom row in Figure 2) is to dynamically determine whether a sentence-video pair is suitable for training or not at the current state. If a pair is considered not suitable for the training at the current state, then this pair does not contribute to the LVC objective. Instead, it contributes to an adversarial loss function L_{ADV}. We interpret how corresponded or informative a sentence-video pair is by the use of some background visual vectors. Acting as inductive biases, the introduced background visual vectors can be seen as some abstracted features that directly capture some background semantics or information which are difficult to interpret visually. The background visual features are randomly initialized and jointly learned with *WAL* . More details refer to the supplementary material. The intuition of the max-pooling layer is to find the largest similarity between the input sentence feature s and all of the background visual features p_{ADV}. To determine whether a sentence-video pair is beneficial for the training, we compare the similarity between the input sentence-video pair p_{LVC} with p_{ADV}. If p_{ADV} is larger than p_{LVC} by a threshold

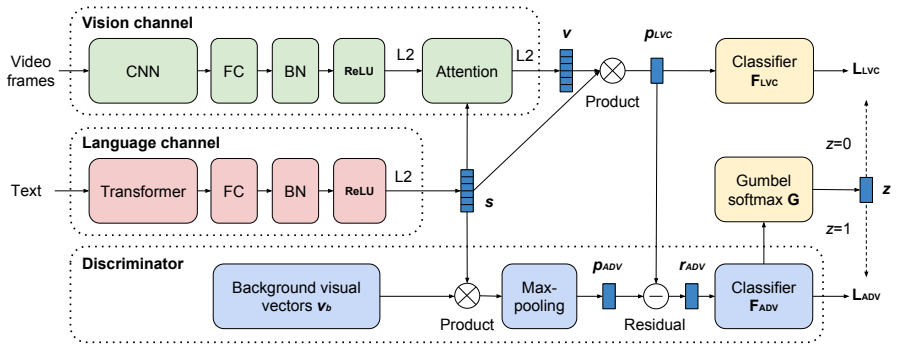

Figure 2: Network architecture of *WAL* . The proposed network architecture consists of three channels. The vision channel and the language channel has an encoder network (ResNet-34 and BERT) and an embedding module that consists of a FC layer, a batch-norm layer and a ReLU layer. The vision channel has an additional attention module which enables the sentence feature s to attend to the related frames in the video. The discriminator consists of a stack of background visual vectors, a max-pooling layer and a classifier. The aim of the discriminator is to dynamically determine whether an input sentence-video pair is suitable for the LVC training at the current state. The binary decision is obtained by sampling a discrete latent variable z using a gumbel-softmax layer. The input pair contributes to the LVC loss if z is 0, otherwise it contributes to an adversarial loss.

(which could be either negative or positive), we then tend to consider the input pair to be not suitable for the training at the current state. However, this input pair might come into play at later training period when the network becomes more capable. The discriminator channel adopts an adversarial training loss for learning, which is explained in Section 2.3.

Relation to existing methods. Common cross-modal retrieval models require human annotations as explicit learning signals. In contrast, the proposed WAL network is designed to learn from noisy data in real-world scenarios. As one of the key contributions in this work, the discriminator enables the network to dynamically discard the noise and learn from the meaningful sentence-video pairs.

2.3 Training Losses

The training loss $L_{overall}$ is the sum of two binary cross-entropy losses, L_{LVC} and L_{ADV}. L_{LVC} corresponds to the LVC task, i.e. determining whether a sentence-video pair is matched. The positive pairs are naturally obtained by time alignment. We then utilize the negative sampling method to sample the negative pairs. The general loss function for the jth pair can be expressed as Equation 1, where y_j denotes the label for the jth training pair, σ is a sigmoid function. y_j can be either 0 (unmatched) or 1 (matched).

$$L(j, f_j) = y_j \log(\sigma(f_j)) + (1 - y_j) \log(1 - \sigma(f_j)). \tag{1}$$

When $z = 0$, we have $L_{\text{LVC}} = L(j, F_{\text{LVC}}(s_j^\top v_j))$, where F_{LVC} is the classifier applied on top of the dot product between the sentence feature s_j and the video feature v_j. When $z = 1$, the loss $L_{\text{ADV}} = L(j, F_{\text{ADV}}(\psi(s_j^\top v_b) - s_j^\top v_j))$ applies instead, where F_{ADV} is the classifier of the discriminator, ψ is max-pooling operation. In particular, we fix y_j for any training pair to be 0 for L_{ADV} to encourage the network to gradually involve more training pairs for L_{LVC}. Hence, L_{ADV} acts as an adversarial loss. The overall loss is therefore the sum of the individual loss over all the pairs.

3 Experiments

Dataset. The collected 'ApartmenTour' dataset contains 2906 YouTube videos in a resolution of 480P. The average length of the videos is about 12 minutes. Each video comes with subtitles provided by either the author or auto-generated by YouTube. In total, there are around 500k sentence-video pairs extracted from this dataset.

Evaluation protocol. We use two standard evaluation metrics for retrieval – mean average precision (mAP) and recall at position k (Rec@k). mAP measures the precision of the predicted ranking averaged over all positions, while rec@k measures how well the positives are retrieved at position k. In this following experiments, we set k to be 5 (and 10 for the second task). Both metrics are multiplied by 100 such that the reported values range from 0 to 100.

Baselines. We compare our networks to 7 baseline methods. Firstly, we report the performance of random guesses as a reference, indicating how much the networks learn from the proposed self-

Table 1: Performance on the 'ApartmenTour' dataset. Video search denotes querying videos using sentences, and vice versa for sentence search. Rec@5 is the recall at position 5. Higher is better for both mAP and Rec@5. 'AT' and 'MS' denote the 'ApartmenTour' dataset and MS-COCO [16].

Network	Training Data	Video Search		Sentence Search	
		mAP	Rec@5	mAP	Rec@5
Random	-	5.2	5.0	5.2	5.0
NIC-BERT	MS	7.8	12.0	6.1	9.0
NIC	MS	19.7	30.0	22.6	37.0
NIC	AT	16.9	25.0	18.8	26.0
NIC	MS+AT	25.7	36.0	25.7	41.0
SCAN LSE+AVG	AT	28.1	41.3	28.2	42.6
PVSE	AT	27.5	40.3	27.1	42.4
WAL	AT	25.0	39.0	26.3	39.8
WAL-att	AT	27.3	41.6	27.9	42.0
WAL-att-adv	AT	**30.1**	**44.2**	**30.0**	**45.4**

supervision task. The next three baselines formulate the retrieval task as a generative modelling problem, and adopt the Neural Image Caption (NIC) architecture introduced in [27]. The three baselines differ in the training procedure. We further propose *NIC-BERT* baseline that adapts the current NIC network for the retrieval task at hand, by mapping both videos and sentences to language features using BERT. Lastly, some strong baselines are implemented. SCAN [15], PVSE [26] are typical supervised learning methods and achieved state-of-the-art performance in the text-image mapping task on MS-COCO.

3.1 Sentence-video bidirectional retrieval

Test set. The dataset for testing consists of 100 positive sentence-video pairs. Each video is of length 2 to 5 seconds. Unlike the training set, the positive pairs in the test and validation set are manually annotated and guaranteed to be *strongly* related. We query the 100 videos using each sentence and average the results for text-video retrieval, and vice versa for the video-sentence search.

Comparison with existing methods. As Table 1 shows, *WAL-att-adv* outperforms all the baselines by a large margin. A surprising observation is that even *WAL* performs on par with the strongest baseline *NIC* which is trained on MS-COCO and fine-tuned on the 'ApartmenTour' dataset. Note that training on MS-COCO is fully supervised and thus provides a very good pre-training. By comparing the three *NIC* baselines, we see that the model trained on MS-COCO can be directly applied to the retrieval task off-the-shelf and achieve competitive results. The performance can be further boosted by fine-tuning on the 'ApartmenTour' dataset as a way to adapt the model to the test set. However, simply training on the 'ApartmenTour' dataset in a generative manner does not provide strong results. This could be due to the noises in the dataset (e.g. loose-correspondence) and incomplete sentences etc. The performance *NIC-BERT* is far from other BERT methods. This is because the generated captions by NIC is quite different from the videos subtitles, e.g. the NIC trained on MS-COCO tends to focus on the person in videos while it is not the case for the subtitles in the 'ApartmenTour' dataset.

SCAN and PVSE are the strongest baselines, as we expected. By combining two complimentary formulations of attention, SCAN performs similar to *WAL-att*, but worse than *WAL-att-adv*. Similarly, PVSE performs on par with SCAN, but is outperformed by *WAL-att-adv*. This further proves the significance of using the discriminator when training networks on a noisy dataset. Moreover, our method is much more memory-efficient (i.e. no need to keep detection features). In general, all the networks trained on the 'ApartmenTour' dataset work significantly better than random guesses. This demonstrates the value of the dataset and the effectiveness of the proposed LVC learning task.

Discriminator. It is impressive to see that the discriminator channel enhances the performance of *WAL-att* significantly, e.g. 27.3 to 30.1 in mAP for image search. This result demonstrates the benefit of the proposed discriminator channel during training. At the beginning of the training, about 60% training pairs contribute to the LVC learning; this number gradually increases and then stabilizes at 95% at the late training epochs, meaning that most of the pairs take part in the LVC training. More details on the behaviour of the discriminator during training refer to the supplementary materials.

4 Conclusion

We define a novel learning task based on the intuition of the language-visual correspondence, and propose a network architecture that can be trained effectively for cross-modal retrieval between sentences and videos, which is shown by both quantitative and qualitative experiments.

5 Appendix

5.1 Additional Details for the Proposed Method

Background visual features. We interpret how corresponded or informative a sentence-video pair is by the use of some background visual features. Acting as inductive biases, the introduced background visual features can be seen as some abstracted features that directly capture some background semantics or information which are difficult to interpret visually. The input sentences without much visual semantics are expected to have high similarities with them, which contribute little to the LVC task. Therefore, we compare the similarity between the input sentence and the background visual features to that between the input sentence and the input video clip, as a way to determine how corresponded or visually informative an input sentence-video pair is. The background visual features are randomly initialized and jointly learned with the WAL network. We use 4 background image features in the discriminator channel, which are initialized by random grouping of the L2-normalized video features v in the training set extracted using the pre-trained ResNet-34 and the subsequent randomly initialized FC layer in the vision channel.

Adversarial training. To enable the network to gradually learn more complex concepts, the LVC objective can be learned in an adversarial manner. To achieve this, we fix the label y_j for any training pair to be 0 in the discriminator for the L_{ADV}. This is because we expect to train the network with more pairs (especially the difficult ones) gradually, and fixing y_j to be 0 reduces the probability of sampling the z to be 1 (i.e. not suitable for training) along the training. This, therefore, can be considered as an adversarial training. Namely, during training, the network tends to sample all the pairs to contribute to the L_{ADV} (i.e. $z = 1$), since in the discriminator the network only needs to always predict 0 (recall that all the labels y for L_{ADV} is fixed to 0).

5.2 Dataset Collection

To obtain the urls of the videos, we use 'apartment tour' and its two extensions (by adding a year or a city name) as the query string query the YouTube search engine. We range the year from 2008 to 2018, and use 28 cities in the world. We manage to download 2906 videos with subtitles. we cut the videos into clips based on the subtitles and corresponding timing. Only the clips with human speech are preserved, while those with no speech or not useful subtitles such as sound indicators are removed. Each clip lasts about 2 to 5 seconds, and 2 frames per second are extracted from each clip. In total, we obtain over 500k pairs.

5.3 Implementation Details

Training details for WAL networks. For each mini-batch during training, we randomly sample the same number of positive and negative sentence-video pairs to avoid the class-imbalance problem. The negative pairs are obtained by first sampling a video, and then randomly sampling a sentence from the 'ApartmenTour' dataset , excluding the original paired sentence. For each epoch, we ensure that every video clip in the dataset is used. During training, we sample a fixed number N_f of frames for each the video for efficiency and as a way of data jittering. If a video has less than N_f frames, some frames are reused. We set $N_F = 5$ in our experiments. Throughout the training, we keep the weights of the pre-trained BERT model and ResNet-34 fixed, as a way to prevent overfitting. Stochastic gradient descent is used to train the network, with weight decay 0.001, momentum 0.9, and an initial learning rate of 0.1; the learning rates are divided by 10 in later epochs. In particular, for the first 50 epochs, the discriminator channel is fixed and its gradients are stopped. This helps to reduce the variance of the gradients at the early training epochs [32]. The whole network is then trained for 20 more epochs.

Data augmentation. The training frames are obtained as follow: first, a crop of a random size (ranging from 0.8 to 1) and a random aspect ratio (between $3/4$ and $4/3$) of the original image is taken. The image crop is then resized to 224×224. A random horizontal flipping and mean subtraction is applied to the image crop. The resultant image is then used as the input to the visual channel of the network. At test time, images are resized so that the smallest dimension is 256 and the central crop of 224×224 is taken.

Training details for NIC baselines. The architecture of NIC baselines consists of a ResNet-152 [9] for image feature extraction and a LSTM for caption generation conditioned on the feature of the input image. The ResNet-152 is pre-trained on ImageNet [5] and fixed for further caption generation training except for batch-norm layers. The training of the NIC baselines is mainly on the sentence decoding part. In detail, the output of a ResNet-152 [9] is first embedded into a feature of size 256 before feeding to the LSTM. The size of the hidden state of the one-layer LSTM is 512. During training for image captioning, the weights of the LSTM, the embedding layer and all the batch-norm layers in the ResNet-152 are updated. During training on the 'ApartmenTour' dataset, for each sentence-video pair, a frame is randomly selected from the video as the image input.

Retrieval method for NIC baselines. At test time, we use the same ranking method as described in [19]. For the sentence-video retrieval, we measure the perplexity of generating the query sentence given each dataset video, which essentially represents how likely each frame in each dataset video can generate the query sentence. The perplexity of a sentence can be expressed as $\log PPL(w_{1:L}|I) = -\frac{1}{L}\sum_{n=1}^{L}\log P(w_n|w_{1:n-1}, I)$, where L is the length of the sentence, $P(w_n|w_{1:n-1}, I)$ is the probability of generating the word w_n given image I and previous words $w_{1:n-1}$. The perplexity of each video is the average perplexity of its frames. Whereas for the video-sentence retrieval, we measure the perplexity of generating each dataset sentence given the query video. Note that the perplexity of each sentence is normalized in the way described in [19] to reduce the bias of the sentence frequency. The training procedure of these three baselines is in the supplementary material.

Retrieval method for NIC-BERT baseline. For the videos, we first use NIC to generate a caption for each frame. In this way, each video is represented by a collection of sentences. We then use BERT to extract language features from the caption of each video frame to form the final video representation. Similarly, the sentences are also represented by their BERT features. As a result, the retrieval can be performed by similarity computation between the BERT features of the videos and the sentences.

Training details for SCAN. To train SCAN [15], we first use a Faster-RCNN [23] (with ResNet-101) trained on Visual Genome [14] to extract the features of 36 ROIs with highest confidence scores in each frame in the dataset. SCAN is then trained on the pre-computed features for 30 epochs with the same λ values as in the paper. For both training and testing SCAN, the similarity score between a sentence and a video is the maximum score between the sentence and all frames.

Training details for PVSE. PVSE [26] is trained on the 'ApartmenTour' dataset with the weights of the diversity loss and the domain discrepancy loss being 0.1 and 0.01. Pre-trained ResNet-152 is used for visual feature extraction and GloVe [22] is used for word embedding. The maximum number of video length of 5. The number of local embeddings used is 3. The network is trained on the 'ApartmenTour' dataset for 70 epochs.

5.4 More Results and Visualization

5.4.1 Ablation Study

We first conduct ablation studies on the discriminator, including varying the number of background visual features (BVF), the sampling layer and the input to the classifier F_{ADV}. We also compare two losses: the cross-entropy loss (introduced in Section 2.3) and the commonly-used triplet loss. The results are shown in Table 2. First, the performance is not very sensitive to the number of BVF in the discriminator, i.e. both models (i.e. BVF=16 and 64) achieve similar performance as the model with BVF=4. In particular, when BVF applies a large number (e.g. 64), we observe that fact that many BVF never fire during training. This can explain the insensitivity of the model to high values of BVF. Second, replacing the gumbel-softmax with the standard softmax also enhances the performance, by comparing *WAL-att-adv* (G=soft.) with *WAL-att*. But the improvement is far smaller than that using the variational inference with a discrete latent variable. This is probably because the training with variational inference and the discrete latent variable is closer to the binary decision that human would make, while the softmax operation is not as intuitive. Third, we can conclude that feeding both p_{ADV} and p_{LVC} (in way of either residual or concatenation) to the classifier F_{ADV} in the discriminator is better than feeding p_{ADV} only. Intuitively, by comparing the two similarity scores (p_{ADV} and p_{LVC}), the discriminator can dynamically make the decisions on whether a sentence-video

Table 2: Ablation study on the sentence-video bidirectional retrieval. 'BVF' denotes the number of background visual features. G is the sampling layer of z and 'Soft.' denotes softmax. 'In.' denotes the input to the classifier F_{ADV} in the discriminator. 'Con.' means the concatenation of p_{LVC} and p_{ADV}.

Network	Video search		Sentence search	
	mAP	Rec@5	mAP	Rec@5
WAL-att (BVF=4)	27.3	41.6	27.9	42.0
WAL-att-adv (BVF=4)	**30.1**	44.2	30.0	45.4
WAL-att-adv (BVF=16)	29.8	44.0	**30.3**	**45.8**
WAL-att-adv (BVF=64)	30.0	44.1	30.2	45.5
WAL-att-adv (**G**=Soft.)	28.8	43.2	28.9	44.2
WAL-att-adv (In.=Con.)	30.0	**44.5**	30.0	45.2
WAL-att-adv (In.=p_{ADV})	28.7	43.0	29.0	44.5
WAL-att (Triplet)	26.7	40.3	26.9	41.2
WAL-att-adv (Triplet)	28.9	43.6	29.2	44.1

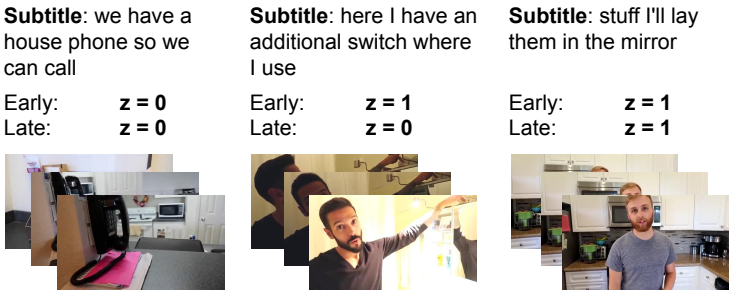

Figure 3: Examples of the discriminator sampling during training. z is shown above each video for the early and late training stages. The first pair is always used for training as the phone is obvious. The second example is not so obvious to determine if they are matched, as it requires the network to infer the relation between the switch and the change of the lighting condition in the video. Hence it is used for the LVC only in the late stage. The last pair is not used for the LVC objective throughout the training, as it is a false positive.

pair is beneficial to the current training. Lastly, we adapt the proposed architecture to train with the triplet loss. Namely, the similarity of each positive sentence-video pair should be larger than those of two negative pairs (i.e. one is the positive clip and the hardest negative sentence in a mini-batch, and vice versa for the other) by at least a margin (0.2). As Table 2 (bottom part) shows, the triplet loss performs on a par with (marginally worse than) the LVC loss. We think that the hardest negative sampling might be too aggressive for such a noisy dataset with many false positive alignments.

5.4.2 Discriminator Behaviour

Figure 3 shows some examples of the sentence-video pairs that are sampled to the adversarial loss (i.e. $z = 1$) during training, meaning that they are considered not suitable for the LVC training. The first two examples are sampled at the early training stages while the other two are sampled at the later training stages. We can see that the network gradually learns from more difficult pairs along the training. However, some pairs are not informative for the learning even in the late training stages.

During the training of the network, we record the number of sentence-video pairs that are sampled to contribute to the language-visual correspondence (LVC) learning task (i.e. the discrete latent variable z is sampled to be 0) in each batch. Furthermore, to reduce the variance, we take the average over all the numbers in each epoch. The resultant numbers are then divided by the batch size, which is 60 in our experiments, to give the percentage of the training pairs that correspond to $z = 0$. Note that the rest of the pairs in each batch contribute to the adversarial loss.

Figure 4 shows how the percentage changes along the training. Note that we only show the percentage of the last 20 epochs of the training, in which the discriminator is trained together with the rest of the network. As we can see in Figure 4, the percentage is around 60% before the training starts (i.e. the 0th epoch). It then reaches 75% after one epoch, meaning that $1/4$ of the sentence-video pairs do not take part in the LVC learning at this point. The percentage increases quickly in the first few epochs to about 90% and then grows slowly to 95% as the training proceeds. This trend matches our

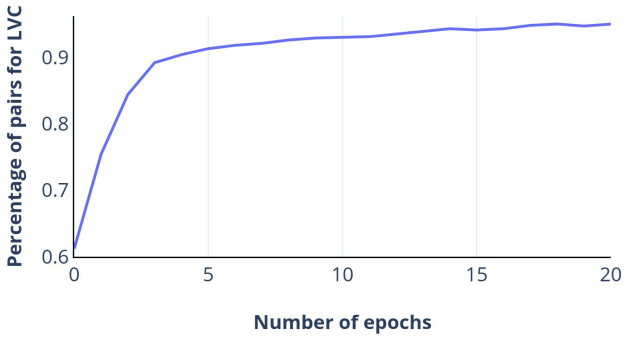

Figure 4: The behaviour of the discriminator during training. The horizontal axis is the number of epochs that the training takes, and the vertical axis denotes the percentage of sentence-video pairs that correspond to $z = 0$. Each epoch covers all the video-sentence pairs in the training set.

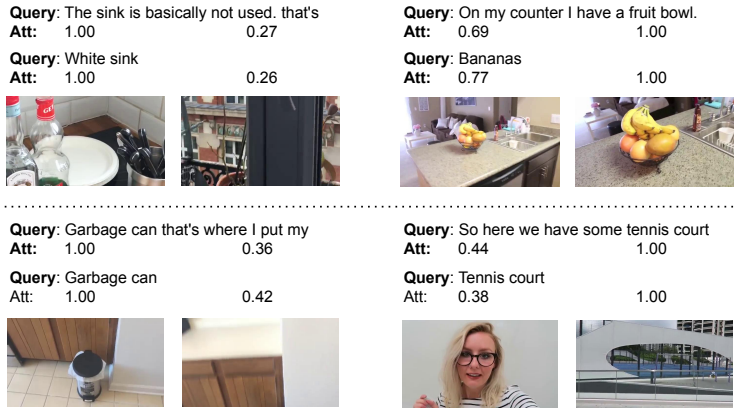

Figure 5: Examples of frame-level attention by query sentences. Each row contains two sentence-video pairs. In each example, the first query is the subtitle corresponding to the video clip (of which only two representative frames are displayed for better viewing), and the second query is simply a phrase or word. The attention score, denoting how much the frame contributes to the final video feature, is shown above each frame. Note that the absolute value is related to the number of frames in the video. For better comparison, the attention scores are normalized such that the larger score is 1.

expectation that the discriminator gradually allows more and more sentence-video pairs to contribute to the LVC loss, by the novel design of an adversarial training setting. Moreover, some pairs are still not used for the LVC learning task at the very end of the training, e.g. the false positive pairs.

5.4.3 Visual Attention Examples

Some examples of the visual attention are shown in Figure 5. It is interesting to find that the attention mechanism also applies to simple query phrases or words (even that do not appear in the original subtitle as in the first row). This proves that the learnt representations do not overfit to the original subtitles.

In Figure 6, we show some more examples of how the input sentence attends to different frames in the video clip based on the relevance between the two modalities. The attention mechanism performs quite well in general. For instance, the first frame on the first row in Figure 6 scores the highest as it contains the plant mentioned in the query sentence, despite that the 3 frames in the video have a very similar background.

There are also some rare cases where the attention mechanism fails. Figure 7 shows two failure examples. In the first example, the network fails to attend to the mattress held by the woman. However, it might because that the mattress is folded and looks very different to what it usually does, i.e. laid on the ground. That, therefore, could make it difficult for the network to recognize the mattress. In

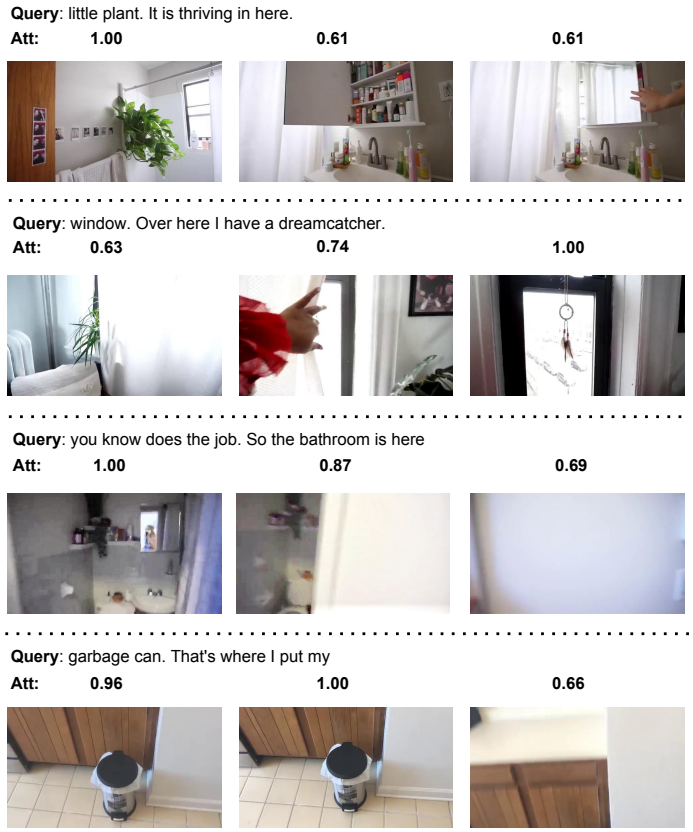

Figure 6: Examples of the sentence-frame attention. Each row contains a sentence-video pair. In each example, the query sentence is the subtitle corresponding to the video clip. For better viewing, only three representative frames are displayed for each video clip. The attention score, denoting how much the frame contributes to the final video feature, is shown above each frame. Note that the absolute value is related to the number of frames in the video. For better comparison, the attention scores are normalized such that the largest score is 1.

Figure 7: Failure examples of the sentence-frame attention. For better comparison, the attention scores are normalized such that the largest score is 1 in each example.

the second case, the network fails to capture the shelving unit in the first frame. We think that it is due to the darkness of the image, and even human find it difficult to recognize.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[3] Andrea Burns, Reuben Tan, Kate Saenko, Stan Sclaroff, and Bryan A Plummer. Language features matter: Effective language representations for vision-language tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7474–7483, 2019.

[4] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10638–10647, 2020.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Chen Gong, Dacheng Tao, Stephen J Maybank, Wei Liu, Guoliang Kang, and Jie Yang. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260, 2016.

[8] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Zhong Ji, Haoran Wang, Jungong Han, and Yanwei Pang. Saliency-guided attention network for image-sentence matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5754–5763, 2019.

[11] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

[12] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014.

[13] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[15] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[17] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[18] Shweta Mahajan, Teresa Botschen, Iryna Gurevych, and Stefan Roth. Joint wasserstein autoencoders for aligning multimodal embeddings. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[19] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014.

[20] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[21] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE international conference on computer vision*, pages 2630–2640, 2019.

[22] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[24] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. Adversarial representation learning for text-to-image matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5814–5824, 2019.

[25] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 200–216, 2018.

[26] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1979–1988, 2019.

[27] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[28] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016.

[29] Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. Camp: Cross-modal adaptive message passing for text-image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5764–5773, 2019.

[30] Jiwei Wei, Xing Xu, Yang Yang, Yanli Ji, Zheng Wang, and Heng Tao Shen. Universal weighting metric learning for cross-modal matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13005–13014, 2020.

[31] Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. Multi-modality cross attention network for image and sentence matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10941–10950, 2020.

[32] Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3732–3741. JMLR. org, 2017.

[33] Huijuan Xu, Kun He, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Text-to-clip video retrieval with early fusion and re-captioning. *arXiv preprint arXiv:1804.05113*, 2018.

[34] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487, 2018.

[35] Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z Li. Context-aware attention network for image-text retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3536–3545, 2020.