Tell me what you see: A zero-shot action recognition method based on natural language descriptions

Valter Estevam[a,c,*], Rayson Laroca[c], Helio Pedrini[b], David Menotti[c]

[a] *Federal Institute of Paraná, Irati-PR, 84500-000, Brazil*
[b] *University of Campinas, Institute of Computing, Campinas-SP, 13083-852, Brazil*
[c] *Federal University of Paraná, Department of Informatics, Curitiba-PR, 81531-970, Brazil*

Abstract

Recently, several approaches have explored the detection and classification of objects in videos to perform Zero-Shot Action Recognition (ZSAR) with remarkable results. In these methods, class-object relationships are used to associate visual patterns with the semantic side information because these relationships also tend to appear in texts. Therefore, word vector methods would reflect them in their latent representations. Inspired by these methods and by video captioning's ability to describe events not only with a set of objects but with contextual information, we propose a method in which video captioning models, called observers, provide different and complementary descriptive sentences. We demonstrate that representing videos with descriptive sentences instead of deep features, in ZSAR, is viable and naturally alleviates the domain adaptation problem, as we reached state-of-the-art (SOTA) performance on the UCF101 dataset and competitive performance on HMDB51 without their training sets. We also demonstrate that word vectors are unsuitable for building the semantic embedding space of our descriptions. Thus, we propose to represent the classes with sentences extracted from documents acquired with search engines on the Internet, without any human evaluation on the quality of descriptions. Lastly, we build a shared semantic space employing BERT-based embedders pre-trained in the paraphrasing task on multiple text datasets. We show that this pre-training is essential for bridging the semantic gap. The projection onto this space is straightforward for both types of information, visual and semantic, because they are sentences, enabling the classification with nearest neighbour rule in this shared space. Our code is available at https://github.com/valterlej/zsarcap.

Keywords: Cross-Dataset, Paraphase Estimation, Video Captioning, Visual GloVe, Zero-Shot Learning

1. Introduction

Human Action Recognition (HAR) is an active research topic in computer vision. Several supervised models have been proposed with an impressive performance in the last years, especially those based on deep learning. At the same time, large-scale datasets containing a massive number of human actions, such as Kinetics-400 [6], Kinetics-700 [5] and ActivityNet [18], have become available. Even in the face of this progress, only a few human actions are mapped, collected and annotated. Hence, retraining state-of-the-art (SOTA) action recognition models is imperative to incorporate new classes, which requires much time, computational resources, energy, and human labor.

Zero-Shot Action Recognition (ZSAR) is a computer vision task that emerges from this problem. In ZSAR, the goal is to recognize examples from unknown human action classes, that is, videos from classes that were not available during the training stage. As we do not have samples from a new class in training, any ZSAR model needs to represent the class labels with semantic information, and the classification is performed with some function, usually learned with from known classes by correlating visual patterns with the label semantic properties.

Traditionally, the videos are represented using spatio-temporal features (e.g., Improved Dense Trajectories (IDT) [50], Convolutional 3D Network (C3D) [48] or Inflated 3D Network (I3D) [6]), and the class labels are represented with attributes or word vectors such as word2vec [33] or Global Vectors (GloVE) [37]. Although this general scheme (deep features ↔ word vec-

*Corresponding author
Email address: valter.junior@ifpr.edu.br (Valter Estevam)

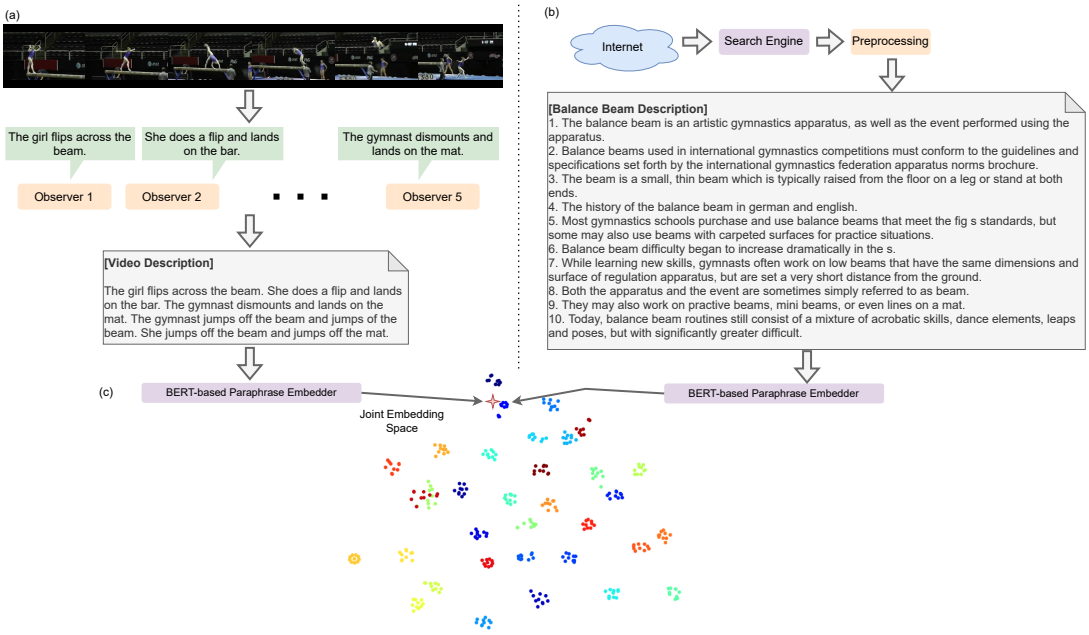


Figure 1: The schematic representation of our ZSAR method. In (a) we show the visual representation procedure. A video is seen by some video captioning systems, called Observers, which produce a video description. In (b) the semantic representation is shown. Using a search engine on the Internet, we collect documents containing textual descriptions for the classes. In this case, the Balance Beam action is preprocessed to select the ten most similar sentences compared to the class name. Finally, in (c), the joint embedding space constructed using a BERT-based paraphrase embedder is used to project both representations in a highly structured semantic space. All information used in the figure comes from real data on the UCF101 dataset.

tors) has become popular for ZSAR, it suffers from a severe domain adaption problem because the learned functions do not transfer well from seen to unseen classes. The main reason is the gap between visual features and semantic features represented with word vectors. For example, different concepts such as *horse riding* and *pommel horse* are prone to appear close into the semantic space, and the absence of complementary information makes it very difficult to discriminate them. It is not surprising that attribute-based methods present higher accuracy than those based on word vectors [11].

As representing classes with a set of attributes is not scalable, some recent approaches have replaced attributes by detecting objects in scenes [21, 31]. This approach works because the visual class-object relationships also exist in texts and are captured in word vectors. Nevertheless, it has some limitations; for example, it can be difficult to distinguish foreground and background objects or provide a proper representation for these object labels in the semantic space. Additionally, the presence of out-of-context objects produces incorrect predictions.

Considering the above discussion, in this work we propose a method in which the goal is to represent the videos and labels with the same modality of information. An intuitive choice is to represent labels and videos with sentences or paragraphs in natural language. In that way, we can produce rich representations for both visual and semantic, and our method is illustrated in Figure 1.

First, we encode the videos using observers that generate a descriptive sentence given an input video, as shown in Figure 1(a). We choose SOTA video captioning methods from [26, 20, 10] pre-trained in the ActivityNet captions dataset (i.e., *without any class label*) due to their remarkable properties, such as (i) using self-attention to concentrate on more important segments in the videos; (ii) storing in their weights video-text relationships; and (iii) producing fluent sentences, which enable us to estimate the similarity between these sentences and the semantic side information using methods for paraphrase identification based on Bidirectional Encoder Representations from Transformers (BERT) encoders pre-trained on large-scale datasets.

We then encode the action labels with texts collected from the Internet through search engines, as illustrated in Figure 1(b). More specifically, we use the descriptions provided in [51] and employ a simple strategy

to select only the sentences most closely related to the action labels. This evaluation is also performed with BERT encoders pre-trained on the paraphrase identification task. As shown in Figure 1(c), these two stages produce a joint embedding space in which a simple nearest neighbor method based on cosine distance achieves remarkable performance.

Our work has some advantages compared to existing methods: (i) the semantic gap due to domain adaptation does not exist or is significantly mitigated when comparing a textual video description with a textual class label description; (ii) a joint representation between visual patterns and texts is encoded in video captioning neural networks, being a natural bridge between these information modalities; (iii) the model is entirely *cross-dataset* and *plug and play*, i.e., we can replace the captioning models with others with better performance or trained on other datasets; we can also replace the BERT-based encoding with an even more accurate encoder with no additional training; and (iv) ideally, no additional training is required to incorporate more classes. It is only necessary to collect texts with descriptions for the labels, which can be automated.

Our contributions are summarized as follows:

- We demonstrate that representing videos with descriptive sentences, automatically learned, instead of deep features is viable and conduct us to the SOTA on the UCF101 dataset in the ZSL scenario;
- We demonstrate that class labels encoded with word vectors are unsuitable for building the semantic embedding space for our approach. Otherwise, we propose representing the classes with sentences extracted from documents acquired with search engines on the Internet without any evaluation of their content;
- We build a shared semantic space employing a BERT-based embedder with a highly accurate pre-trained model for the paraphrasing task. The projection onto this space is straightforward for both types of information;
- Finally, our experimental evaluation demonstrated that the main performance limitation is the current state of the art on video captioning, which can be considerably improved in the coming years.

2. Related Work

The central problem in ZSAR is how to bridge the gap between what the model is seeing and the semantic knowledge it has. As shown in [11], existing methods based on attributes manually annotated by humans reached greater accuracy than raw deep representations. However, video annotation is not scalable, and different approaches have been proposed to represent videos with automatically detected attributes, usually the presence or absence of objects, classified by knowledge transfer from large-scale datasets. Recently, the use of textual representations to learn joint representations has been proposed with promising performance. In the following subsections, we introduce some relevant approaches for these strategies.

2.1. Object Representations for ZSAR

Guadarrama et al. [17] proposed an approach based on hierarchical semantic models for subjects, objects, and verbs. They employed object detectors associating the predictions with their corresponding leaves in the hierarchies. Information from objects and subjects is combined and fed into a non-linear Support Vector Machine (SVM). On the other hand, Jain et al. [21] used the estimated probability of detected objects as prior knowledge and estimated an affinity between an object class and an acting class. This information was used to compute the semantic description of an action class as a function of the set of predicted objects.

Wu et al. [53] proposed generating an intermediate space containing the relationships among objects, scenes, and actions. They employed a semantic fusion network on three streams: global low-level Convolutional Neural Network (CNN) (e.g., from a VGG19 trained on ImageNet); object features in frames (e.g., from VGG19 trained on a subset of 20,574 objects); and features of scenes (e.g., from a VGG16 trained on the Places205 dataset). The correlation between objects/scenes and video classes is mined from the visualization of the network by salience maps producing a matrix with the probability that each pair (object, scene) is related to an action. Mettes and Snoek [31], on the other hand, focused on the spatial relationship between actors and objects. They proposed a method based on spatial-aware object embeddings computed from interactions between actors and local objects in sequential frames using a pre-trained Faster R-CNN model on the MS-COCO dataset. Segments with actor-local object interaction were called action tubes, and these tubes are distinguished among different videos using global object classifiers through the GoogleLeNet network. The video class is determined as the class with the highest combined score between video tube embeddings and global classifiers. Their semantic information

is given by cosine distance of actions and objects taken word2vec representations.

Gao et al. [12] learned the relationship between actions and objects in a two-stream configuration. In the first stream, they learned classifiers on graph models constructed with ConceptNet5.5 [45], where the concepts are represented with word vectors. The second stream used the visual representations of objects (with the methods used in [21] and [31]) to learn the graphs. The classifiers are learned during training and optimized for seen categories. Hence, in testing, the classifiers of unseen categories (i.e., from the first stream) are used to classify the object features of test videos (i.e., from the second stream). This method is the inspiration for the approach of Ghosh et al. [14], which feeds knowledge graphs to a Graph Convolutional Network (GCN), aiming to minimize the Mean Squared Error (MSE) between the final classifier layer weights (GCN) with the classifier layer weights from I3D.

Finally, Kim et al. [22] proposed generating semantic embedding spaces based on dynamic attributes signatures. They showed that dynamic attributes are preferable to static ones for modeling actions due to the lack of temporal information. Thus, they constructed finite state machines over the static annotations provided in the UCF101 and Olympic Sports datasets describing the presence and the transitions between these states. These patterns are action signatures used to perform the ZSAR classification.

Our method explores the ability of video captioning to identify objects in scenes inferred by their context and by sentence annotations. Additionally, we employ the I3D model as a deep representation, and this model incorporates the weights of an Inception-V1 model pre-trained on ImageNet [6].

2.2. Text Representations for ZSAR

Zhang et al. [57] proposed an improved model for learning visual and textual alignments. Tipically, these approaches take a set of paragraphs, represented as a sequence of words, and feed it into an encoder to obtain a paragraph embedding. Similarly, a set of short clips composed of a few frames is fed to an encoder to obtain a video embedding. These embeddings are updated with a loss function at a high level (e.g., cosine distance). Their method proposes a mid-level alignment where paragraphs are aligned to videos and sentences are aligned to short clips. The quality of the intermediate encoding is improved by using decoding networks to evaluate reconstruction errors.

Piergiovanni and Ryoo [38] also developed a method to learn an intermediate representation for both videos and texts based on an encoder-decoder approach. In their method, there are two encoder-decoder pairs: (video-encoder, video-decoder) and (text-encoder, text-decoder). The first encoder takes a video and produces an intermediate space, and the first decoder reconstructs the video given the intermediate representation. The same occurs with text. Four loss functions were proposed to handle the learning with paired and unpaired data. The classification is performed by the nearest neighbor rule between each video representation and its text representation in the intermediate space.

Recently, Chen and Huang [8] proposed a method combining object detection and textual information. They observed that only word vector representation is insufficient to provide information for objects detected in the videos. Then, they used the object label to retrieve their WordNet description as an object concept description. Additionally, they proposed a combination of Wikipedia and dictionary data to compose action class descriptions using human supervision in this task. Hence, they could identify objects in videos and provide a representation based on their concepts. Although well succeeded, their method requires the presence of visual representation in the ZSAR classification step.

Our method is also based on textual descriptions, but it has several differences: (i) we use methods that predict descriptions word by word and consider the visual information and the previously predicted words. A clear advantage of this strategy is to ignore objects out of context; (ii) our method does not require any class label annotation nor to train the ZSAR classifier; (iii) our strategy for semantic side representation does not require human supervision at the level of sentences; it requires only a document from the Internet with a general description; and (iv) as we have good descriptions, paraphrase identification methods pre-trained on millions, or even billions of sentences, can be employed without the need for fine-tuning.

3. Methodology

In this section, we describe in detail our methodology, which is illustrated in Figure 1.

3.1. Problem Definition

The goal of ZSAR is to classify samples belonging to a set of unseen action categories $\mathcal{Y}_u = y_1, ..., y_{u_n}$ (i.e., never seen before by the model) given a set of seen categories $\mathcal{Y}_s = y_1, ..., y_{s_n}$ as the training set. The problem is named ZSAR only if the following restriction is respected:

$$\mathcal{Y}_u \cap \mathcal{Y}_s = \emptyset \quad (1)$$

Our classification consists of mapping both video and semantic information (i.e., class description) into a joint embedding space. Then, the classification is performed with a nearest neighbor rule under some similarity function, such as

$$y_{pred} = \underset{y_{prot} \in \mathcal{Y}_{u_{prots}}}{\arg\max} \; Sim(Emb(y_{prot}), Emb(Obs(v))) \quad (2)$$

in which *Sim* is the cosine similarity; v is a video, $Obs(\cdot) = [Ob_1(\cdot), ...Ob_o(\cdot)]$; $[\cdot]$ is a concatenation operator and $Ob(\cdot)$ is a video sentence description from each of the o observers (i.e., video captioning methods) (see details in Section 3.2); y_{prot} is a sentence from a large textual description for each class obtained with the procedure described in Section 3.3; finally, $Emb(\cdot)$ is a sentence embedding function described in Section 3.4. Our method, as mentioned previously, does not use the training set because the benchmark datasets do not provide annotated sentences for their videos.

3.2. Video Representation

Our goal is to predict a sentence given a video (using visual and audio information when available). As video captioning is an area of computer vision responsible for study models with this ability, we choose two SOTA architectures that could be used with the same set of features: Transformer [26] (using the original transformer implementation from [49]), and Bi-Modal Transformer [20]. Figure 2 shows a diagram illustrating both models.

Transformer: First, given a video V, the observer takes a set of n_c visual features $V_f = \{v_{f_1}, ..., v_{f_{n_c}}\}$, one per each frame stack, and a set of m words $Y = \{y_1, ..., y_m\}$ to estimate the conditional probability of an output sequence given an input sequence.

We encode v_{f_c}, where $1 \leq c \leq n_c$ as

$$v_{f_c} = V_E(v_c)\,, \quad (3)$$

where $V_E(\cdot)$ yields a deep representation given by an off-the-shelf convolutional network, and v_c is the c-th frame stack for the video V. The video features (Equation 3) are fed all at once to the transformer encoder in which a learned continuous representation is passed to a decoder to generate a sequence of symbols Y from the language vocabulary.

The Transformer requires information on the position of each feature, and a usual strategy is to compute a positional encoding with sine and cosine at different frequencies as

$$\begin{aligned} PE_{(pos,2i)} &= \sin(pos/10000^{2i/d_{model}}), \\ PE_{(pos,2i+1)} &= \cos(pos/10000^{2i/d_{model}}), \end{aligned} \quad (4)$$

where pos is the position of the visual feature in the input sequence, $0 \leq i < d_{model}$ and d_{model} is a parameter defining the internal embedding dimension in the transformer. Following, a multi-head attention layer process these representations with scaled dot-product attention defined in terms of queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$) as

$$Att(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}), \quad (5)$$

and the multi-head attention layer is the concatenation of several heads (1 to h) of attention applied to the input projections (computed with dense layers) as

$$MHAtt(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [head_1, ..., head_h]W^0\,, \quad (6)$$

where $head_i = Att(\mathbf{Q}W_i^{\mathbf{Q}}, \mathbf{K}W_i^K, \mathbf{V}W_i^{\mathbf{V}})$ and [ ] is a concatenation operator. The key insight on Transformer is the self-attention, which takes $\mathbf{Q} = \mathbf{K} = \mathbf{V} = V_f^{PE}$, resulting in

$$\begin{aligned} V_f^{self-att} = [&Att(V_f^{PE}W_i^{V_f^{PE}}, V_f^{PE}W_i^{V_f^{PE}}, V_f^{PE}W_i^{V_f^{PE}}), \\ &..., Att(V_f^{PE}W_h^{V_f^{PE}}, V_f^{PE}W_h^{V_f^{PE}}, V_f^{PE}W_h^{V_f^{PE}})]. \end{aligned} \quad (7)$$

The latent feature from the encoder is given by a fully connected feed-forward network $FFN(\cdot)$ applied to each position separately and identically, defined as

$$FFN(u) = \max(0, uW_1 + b_1)W_2 + b_2\,, \quad (8)$$

resulting in V_f^{FFN}, which is a rich video representation based on self-attention used in the decoder layer.

The decoder layer receives words and feeds an embedding layer $E(\cdot)$, computing the position with Equation 4 resulting in W^{PE}. This representation is fed to the multi-head self-attention layer to compute an internal representation based on self-attention applied on word sequence, resulting in $W^{self-att}$.

Then, we compute the relationship between video and sentence by feeding the encoder-decoder attention layer, resulting in an attention on the words given the visual encoding as

$$W^{VisAtt} = MHAtt(W^{self-att}, V_f^{FFN}, V_f^{FFN}). \quad (9)$$

Finally, W^{VisAtt} feeds an $FFN(\cdot)$ and, then, a generator $G(\cdot)$ composed of a fully connected layer and a softmax layer is responsible for learning the predictions over the vocabulary distribution probability. This model is highly efficient in modeling visual-textual relationships.

Bi-Modal Transformer (BMT): The second architecture employed is BMT. Considering the encoder, this

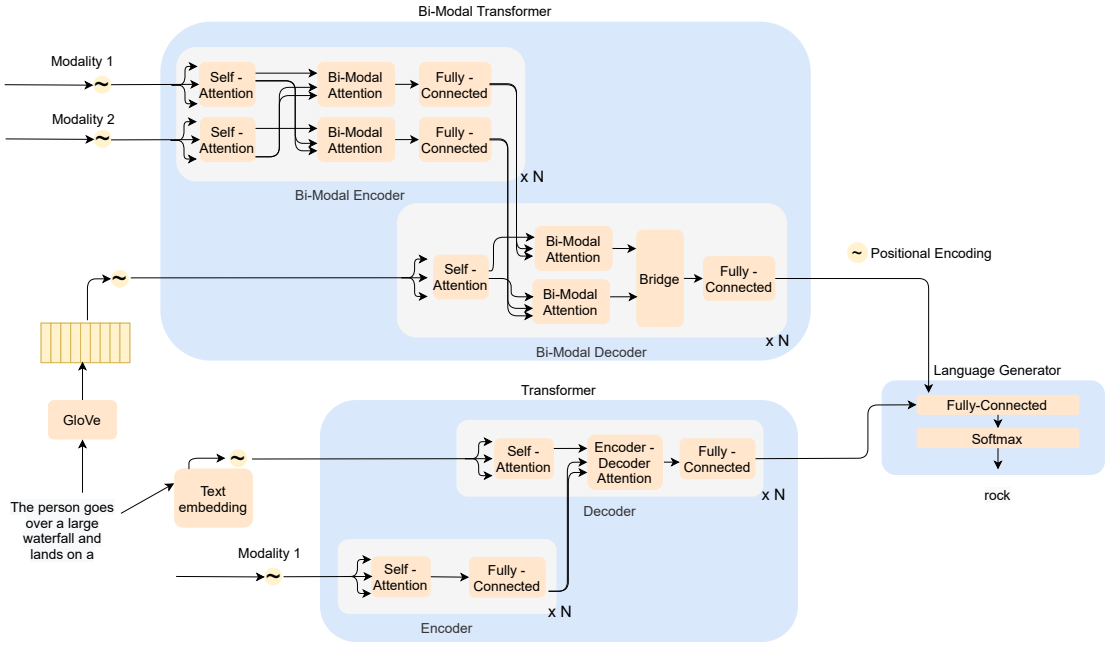


Figure 2: Overview of the captioning architectures showing the Bi-Modal Transformer and Transformer layers with their inputs and the language generation module. Adapted from: Estevam et al. [11].

transformer has two differences from the Transformer encoder. It takes two streams, visual V_f and audio A [20] or semantic Sm [10], separately. We denote this second stream as ASm (i.e., audio or semantic). The encoder has three sub-layers: self-attention (Equation 5), producing $V_f^{self-att}$ and $ASm^{self-att}$; bi-modal attention, i.e.,

$$V_f^{ASm-att} = MHAtt(V_f^{self}, ASm^{self}, ASm^{self}), \quad (10)$$

$$ASm^{Vis-att} = MHAtt(ASm^{self}, V_f^{self}, V_f^{self}), \quad (11)$$

and a fully connected layer $FFN(\cdot)$ for each modality attention, producing $V_{ASm-att}^{FNN}$ and ASm_{v-att}^{FNN} used in the bi-modal attention units on the decoder.

Considering the bi-modal decoder, a $W^{self-att}$ is obtained with Equation 6. Afterward, the bi-modal attention is computed as

$$W^{ASm-att} = MHAtt(W^{self-att}, ASm_{v-att}^{FNN}, ASm_{v-att}^{FNN}), \quad (12)$$

and

$$W^{V-att} = MHAtt(W^{self-att}, V_{ASm-att}^{FNN}, V_{ASm-att}^{FNN}). \quad (13)$$

The bridge is a fully connected layer on the concatenated output of bi-modal attentions, which are enriched features through attention on the combination of two video modalities (e.g., visual and audio), computed as

$$W^{\text{FFN}} = \text{FFN}([W^{Sm-att}, W^{V-att}]). \quad (14)$$

The output of the bridge is passed through another FFN and then to the generator $G(\cdot)$. This means that the encoder parameters are learned conditioning them to the sentence output quality.

We compute the semantic descriptor from [10] strictly following the model and training procedures. The mathematical details can be found in the original paper.

3.3. Class Label Representation

We take a dataset with documents collected on the Internet containing a textual description for each class. Hence, for each class, we have a set of prototype sentences $S_{prot} = \{s_{p_1}, s_{p_2}, ..., s_{p_q}\}$ obtained by splitting the paragraphs.

We employ simple but effective selection criteria: (i) to filter the sentences with a minimum number of words; (ii) to compute dense representations for all the sentences and the class label using the Sentence-BERT (SBERT) [40] model; (iii) to compute the cosine similarity between the dense representations of the class label and the sentences; and (iv) to select a maximum number of sentences ordered by the highest similarity.

The joint embedding space used for ZSAR is composed of representations for video and prototype sentences computed with the SBERT model. The details are provided in the following section.

3.4. Sentence Embedding

We propose to encode information at the level of sentences and not words. For this task, we use the SBERT model from [40]. It is an improved BERT [9] model that drastically reduces the computational cost for acquiring BERT embeddings by feeding a Siamese network, containing two BERT models, with one sentence per branch, dispensing with the special token [SEP]. The model architecture is shown in Figure 3.

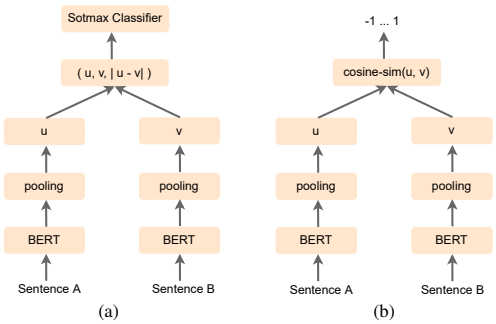


Figure 3: SBERT architecture from Reimers and Gurevych [40]. In (a) is shown the classification objective function, and in (b) the architecture used at the inference or regression tasks.

BERT or RoBERTa models are fine-tuned on large-scale textual similarity datasets. If the dataset requires classification, the objective function is described as

$$o = softmax(W_t[u, v, |u - v|]) \tag{15}$$

where $[\cdot]$ is the concatenation operator, $|u - v|$ is an element-wise subtraction, $W_t \in \mathbb{R}^{3n \times k}$ is the trainable weights, n is the dimension of sentence embeddings, and k is the number of labels. The model optimizes the cross-entropy loss. On the other hand, if the dataset requires regression, the cosine similarity between two sentence embeddings u and v is computed, and the loss function is the mean squared error.

The model can also be optimized using a triplet objective function. Taking an anchor sentence a, a positive sentence p, and a negative sentence n, the triplet loss tunes the network so that the distance between a and p is smaller than the distance between a and n, that is, minimizing the following equation

$$max(\|s_a - s_p\| - \|s_a - s_n\| + \epsilon, 0), \tag{16}$$

where s_a, s_p, and s_n are sentence embeddings, $\|\cdot\|$ is a distance metric and ϵ is a margin ensuring that s_p is at least ϵ closer to s_a than s_n.

Our interest is in the vector u (see Figure 3), after the fine-tuning, computed as the mean of all outputs instead only output for [CLS], as occurs in BERT. For details on BERT or RoBERTa see [9] and [29], respectively.

4. Experiments

In this section, we introduce the datasets and protocols, the implementation details, and the results. We also include an extensive ablation study organized as a set of questions and answers (*Q&A*).

4.1. Datasets and Protocol

Our observers were trained using the ActivityNet Captions dataset [23], which consists of 10,024 training, 4,926 validation, and 5,044 testing videos collected from YouTube. The videos are annotated with start and end points for events, and a sentence is provided for each annotation totaling approximately 36K pairs of event-sentence. The sentences have an average length of 16.5 words and describe around 36s of their videos. It is important to highlight that no action label from ActivityNet is used during the training of the video observers.

For testing, we employ the popular benchmarks HMDB51 [24] and UCF101 [44]. The former is composed of 6,766 videos from 51 classes, with an average duration of 3.2s; the frame height is scaled to 240, and the frame rate is converted to 30 frames per second (FPS). The latter comprises 13,320 videos from 101 action classes with frame resolution standardized to 25 FPS and 320×240 pixels. The average duration of the videos is 7.2s. Performance is evaluated through the accuracy metric.

Providing a fair evaluation of ZSAR models using these datasets is not straightforward due to the nature of the visual feature extractors and the datasets used for training them. For example, if a ZSAR model uses the I3D network, pre-trained on Kinetics400 [6], there are overlaps between the set of classes from Kinetics400 and the set of classes from HMDB51 and UCF101. This overlap imposes the removal of these classes from the ZSAR test set to preserve the Zero-Shot Learning (ZSL) premise (i.e., the disjunction between training and testing class sets). However, these overlaps are often challenging to recognize due to differences in class names and the visual and semantic similarity between certain classes, as pointed out in [11, 3, 42, 8, 16].

Taking this into account, we adopt the TruZe evaluation protocol [16] on UCF101 and HMDB51 in which the testing split is generated with the following guidelines: (i) to discard exact matches (e.g., archery); (ii) to discard matches that can be either superset or subset (e.g., cricket shot and cricket bowling (UCF101) and playing cricket (Kinetics400)); and (iii) to discard matches that predict the same visual and semantic match (e.g., apply eye makeup (UCF101) and filling eyebrows (Kinetics400)). The result is a configu-

ration with 29/22 (train/test) and 67/34 classes for the HMDB51 and UCF101 datasets, respectively. As our model does not require these training sets (i.e., it is cross-dataset), we take into consideration only the testing sets (i.e., 0/22 and 0/34): **UCF101** - apply lipstick, balance beam, baseball pitch, billiards, blow dry hair, cutting in kitchen, fencing, field hockey penalty, front crawl, hammering, handstand pushups, handstand walking, horse race, ice dancing, jumping jack, military parade, mixing, nunchucks, parallel bars, pizza tossing, playing daf, playing dhol, playing sitar, playing tabla, pommel horse, punch, rafting, rowing, still rings, sumo wrestling, table tennis shot, uneven bars, wall pushups, and yo yo; **HMDB51** - chew, climb stairs, draw sword, fall floor, fencing, flic flac, handstand, hit, jump, kick, pick, pour, run, sit, shoot gun, smile, stand, sword exercise, talk, turn, walk, and wave.

4.2. Implementation Details

We compute features as shown in Figure 4. For all videos, we extract features from all datasets using the I3D network with its two streams, RGB and Optical Flow, in videos with 25 FPS. We follow the authors' recommendations for re-scaling (224×224 pixels) but replace the TV-L1 [34] optical flow algorithm for the PWC-Net [47], as it is much faster[1].

For each video, we extract one feature with stacks of 24 frames and steps of 24 frames (i.e., 0.96 features per second). The audio features are extracted with the VGGish model [19] pre-trained on AudioSet [13]. We follow the default configuration. Considering that the videos on the HMDB51 dataset do not have the audio signal and that around 50% of the videos from UCF101 have this information, we compute the Visual GloVe features [10] from RGB stream of I3D, which is a simple and effective feature to replace the audio stream in the BMT model and to enrich the Transformer model input. Finally, we get four features: VisGloVe, i3DVisGloVe, i3D, and VGGish (see Figure 4(a)). With these features, we fed two architectures for video captioning (i.e., Transformer and BMT) which allowed us to generate 5 distinct observers. Figure 4(b) shows the configuration of each observer (architecture and inputs).

The Transformer and BMT models are trained up to 60 epochs employing early stopping if the Meteor score [2] stays unchanged for 10 epochs. The loss function adopted is the Kullback-Leibler Divergence with label smoothing and masking. Dropout is used to prevent overfitting with a rate of 0.1. Additionally, we monitor the Bleu@3 and Bleu@4 scores [36] to allow evaluating the quality of the sentences produced during the training stage. The VisualGlove features are computed with a vocabulary of 1,000 visual words (learned with clustering), a context of 25 words ($\approx$ 24s), and a dimension of 128. The training is performed until 1,500 epochs with early stopping of 100 without improvements in the cost function.

The adoption of multiple observers is motivated by the intuition that different humans will produce different sentences given a sample video. Although different, these sentences tend to be complementary to each other. As our results show, this scheme is highly efficient in improving the video representation, which is reflected in the increase of ZSAR accuracy considering multiple sentences.

We build the semantic space with Sentence-BERT encoders [40], namely, the *paraphrase-distilroberta-base-v2*[2] model [41]. We use the textual descriptions provided by Wang and Chen [51][3] as side information. The texts are processed using the NLTK[4] package for splitting paragraphs into sentences and the *contractions*[5] package to expand contractions (e.g., "isn't" to "is not"). We follow the procedure described in Section 3.3 by selecting sentences with a minimum of 10 words and up to 10 sentences per class and taking the nearest sentence encodings compared to the label encoding. We employ the cosine distance as the similarity measure. The sentences from the observers are concatenated and processed with *paraphrase-distilroberta-base-v2* and a Nearest Neighbor algorithm from *scikit learn*[6] is adopted as the ZSAR classifier.

4.3. Selected Benchmarks and Evaluation

We selected two generic ZSL models and four SOTA ZSAR methods for comparison, briefly described in this section.

Latem [54] is a direct projection onto semantic space method in which a piece-wise linear compatibility function is used to understand the visual-semantic embedding relationships. SYNC [7] generates a weighted

[1]The code used for feature extraction is available in `https://github.com/v-iashin/video_features`

[2]Trained on the following datasets: AllNLI, sentence-compression, SimpleWiki, altlex, msmarco-triplets, quora_duplicates, coco_captions, flickr30k_captions, yahoo_answers_title_question, S2ORC_citation_pairs, stackexchange_duplicate_questions, wiki-atomic-edits.

[3]The data is available at `https://staff.cs.manchester.ac.uk/~kechen/ASRHAR/`

[4]`https://www.nltk.org/`

[5]`https://pypi.org/project/contractions/`

[6]`https://scikit-learn.org/`

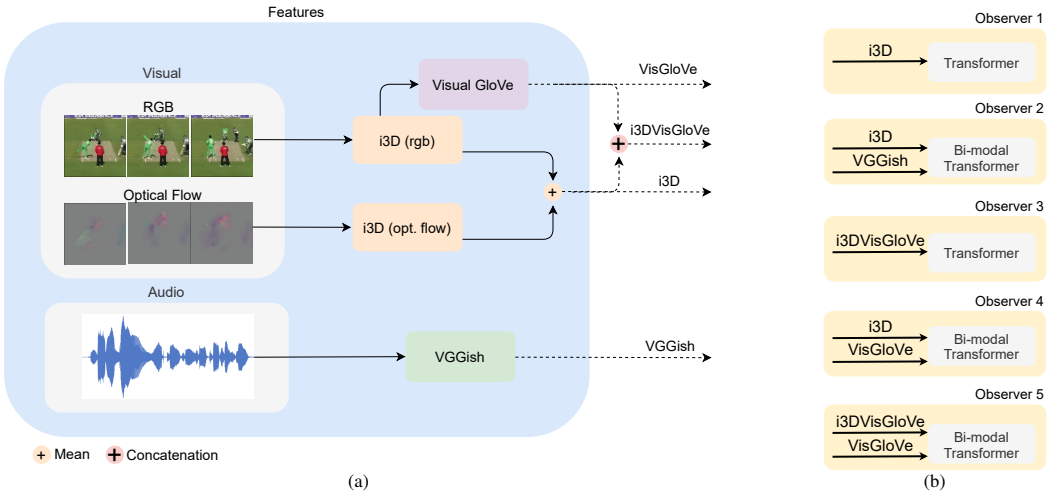


Figure 4: Features and observers. In (a) is shown features computed from visual and audio streams, and in (b) the observers architecture and their respective input features.

graph with synthesized classes that ensure the alignment between semantic embedding space and the classifier space by minimizing the distortion error. BiDiLEL [52] learns two projection functions for projecting visual and semantic spaces onto a shared embedding space to preserve the relationship between them. OutDist [30] learns a visual feature synthesizer given the semantics and an out-of-distribution detector to distinguish generated features from seen ones. E2E [3] learns a CNN to generate visual features for unseen classes by training (in an end-to-end manner) this model with a combined dataset taking classes from Kinetics400 and overlapping classes of UCF101 and HMDB51. Finally, CLASTER [15] applies reinforcement learning on the clustering of visual-semantic embeddings.

Table 1: Observer accuracy for the UCF101 and HMDB51 datasets taking the 34 and 22 testing classes from TruZe, respectively. Note that no training classes were used to train the models.

OB1	OB2	OB3	OB4	OB5	HMDB51	UCF101
✓					14.4	38.6
	✓				–	37.2
		✓			13.5	34.6
			✓		12.7	30.9
				✓	10.6	35.3
✓		✓			14.8	44.9
✓		✓	✓		14.2	47.3
✓		✓	✓	✓	14.5	48.0
✓	✓				–	46.5
✓	✓	✓			–	48.9
✓	✓	✓	✓		–	48.9
✓	✓	✓	✓	✓	–	**49.1**

4.4. Results

In Table 1, we show the ZSAR performance considering each observer individually, as well as some combinations of them. There is a huge difference in the accuracy rates achieved in the HMDB51 and UCF101 datasets, taking the same captioning models. Therefore, we discuss the results for each dataset separately.

In the UCF101 dataset, we observe that combining multiple observers has a considerable impact on performance. The complete model is 27% (i.e., 49.1/38.6) more accurate than the best observer individually. This property is a clear advantage of our model since new observers can be included later, thus improving overall performance. Another interesting case is the inclusion of OB2, which uses i3D and VGGish (see Figure 4(b)). As mentioned earlier, approximately 50% of the videos have audio signal. However, this observer has a high individual performance and increases the final result by 2.3% (i.e., 49.1/48) compared to the best performance without it.

Regarding the HMDB51 dataset, we believe that it is a challenging dataset for our approach mainly due to the short length of the videos (i.e., just 3.2 seconds on average), which implies short stacks of features that nullify the benefits from self and multi-modal attention mechanisms. This is evidenced by the fact that observers with different inputs do not learn better descriptions, as with the UCF101 dataset. In order to investigate this hypothesis, we extract features by reducing the frame stack length to 10 and 16 frames, corresponding to one i3D feature at 0.40 and 0.64 seconds, respectively. Table 2

shows the results acquired with these features taking the same pre-trained models used in Table 1. Notably, the performance is improved by 38%, considering the best cases from both tables (20.4/14.8). We note that, for this particular dataset, it is better to consider only observers based on Transformer models. This can be explained based on the characteristics of Visual GloVe features, which encode co-occurrence of visual patterns in complex events with long duration (one minute on average with a window of 24s) [10]. Hence, BMT-based observers are not suitable for this dataset. On the other hand, Visual GloVe proves to be useful as a feature enricher with Transformer (observer OB3), as evidenced by the increase of 7% (OB1+OB3) compared to the I3D version alone (observer OB1) (i.e., 20.4/19.1).

Table 2: Observer accuracy for the HMDB51 dataset taking 22 testing classes from TruZe. We changed the number of frames used to compute visual features (from 24 to 10/16).

10	16	OB1	OB3	OB4	OB5	HMDB51
✓		✓				19.1
✓			✓			17.8
✓		✓	✓			**20.4**
✓				✓		14.9
✓					✓	14.3
✓		✓	✓	✓	✓	19.1
	✓	✓				19.2
	✓		✓			16.6
	✓	✓	✓			19.2
	✓			✓		16.5
	✓				✓	15.7
	✓	✓	✓	✓	✓	19.1

Finally, Table 3 shows the comparison with the selected baselines. As can be seen, the proposed method achieves state-of-the-art performance on the UCF101, even without using the 67 classes from the training set. Despite the issues regarding our method and the HMDB51 dataset, we obtain a remarkable performance.

4.5. Ablation Studies

Here, we present a set of questions and answers *Q&A* to demonstrate the effectiveness of our approach. In all experiments, we use the same observers from the results shown in Table 3.

4.5.1. Is human involvement necessary for action class representation?

Chen and Huang [8] introduced a method based on Elaborative Descriptions (ED) (i.e., a concatenation of class name and its sentence-based definition). These descriptions were constructed by crawling candidate sentences from Wikipedia and dictionaries using action names as queries. Afterward, annotators were asked to select and modify a minimum set of sentences. Table 4 compares the ZSAR performance considering four scenarios: only class label, ED, Ours + ED, and only Ours.

Table 3: SOTA comparison under the TruZe protocol [16]. tr/te = train/test split configuration; Acc = accuracy.

	HMDB51		UCF101	
	tr/te	Acc.	tr/te	Acc.
Latem [54]	29/22	9.4	67/34	15.9
SYNC [7]	29/22	11.6	67/34	15.0
BiDiLEL [52]	29/22	10.5	67/34	16.0
OutDist [30]	29/22	21.7	67/34	23.4
E2E [3]	29/22	31.5	67/34	45.2
CLASTER [15]	**29/22**	**33.2**	67/34	45.3
Ours	0/22	20.4	**0/34**	**49.1**

The results in both datasets show that the proposed pre-processing method achieves a higher accuracy compared to others. Although ED reached impressive results in [8], it did not prove efficient for adoption with our method, in which the joint embedding (visual and semantic) is based exclusively on transfer learning from the Natural Language Processing (NLP) domain. We believe this occurs due to the lack of fine-tuning with the descriptions of training classes in our method.

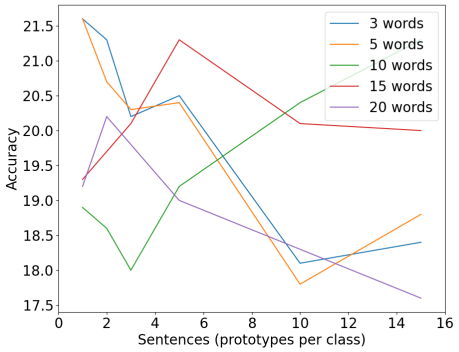
Table 4: ZSAR performance on the HMDB51 and UCF101 datasets considering different semantic information modalities. All experiments were conducted on the TruZe protocol.

	HMDB51	UCF101
Baseline (only label)	19.5	36.6
Elaborative Descriptions [8]	14.1	32.5
Ours + Elaborative Descriptions	19.4	43.9
Ours	**20.4**	**49.1**

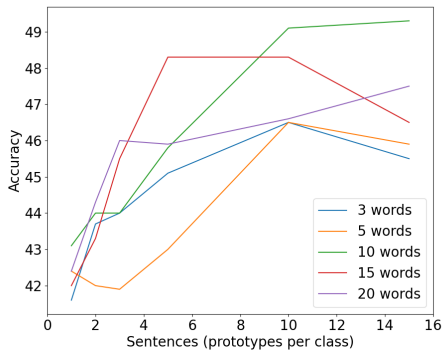
Considering these results, we propose the following question:

4.5.2. How many sentences are required, and how is the ideal minimum length to represent class labels?

Figures 5(a) and 5(b) show the accuracy considering a minimum length of 3, 5, 10, 15 and 20 words per sentence for HMDB51 and UCF101, respectively. We change the maximum number of sentences per class (i.e., the number of prototypes in semantic space for each class) for each minimum length value.



(a)



(b)

Figure 5: ZSAR performance for different configurations of the prototypes. We change the maximum sentences per class, taking 3, 5, 10, 15, and 20 minimum words per sentence. (a) shows the results from HMDB51 and (b) from UCF101.

The graphs clearly show the need to balance the number of words and the number of sentences. There is a tendency for decreasing performance as more sentences are considered in HMDB51 and, conversely, an increasing in UCF101. Using short sentences, we inevitably select loose sentences containing the class label (i.e., section titles or image labels in HTML pages), thus failing to capture the semantic context. On the other hand, when selecting long sentences with 15 or 20 words, we restrict the model to long explanations, failing to capture the immediate context of the class label. Therefore, our configuration (minimum of 10 words and up to 10 sentences) is a good trade-off between a minimum set of words and a maximum number of sentences in both datasets.

Additionally, the graph from Figure 5(a) illustrates another aspect of why HMDB51 is so challenging for our method. The configurations with 3 or 5 words and only one sentence present the better performance, possibly because some actions in this dataset (e.g., chew, pick, turn and wave) are semantically represented with a dictionary-style description (i.e., short and precise descriptions). This behavior is also evidenced in Table 4.

4.5.3. Should we represent the class labels with separated sentences or with a paragraph?

We can represent each class label with sentences or with a paragraph composed of the same sentences concatenated. Table 5 shows the results taking only the class label (i.e., one prototype per class, a single paragraph (i.e., one prototype per class), or ten sentences (i.e., ten prototypes per class). Using sentences proves to be more accurate than the other options in both datasets. This characteristic is a remarkable aspect of our approach because other ZSAR methods always consider only one prototype. Additionally, the paragraph representation proves to be better than the label name for our approach on UCF101. Indeed, the label name is insufficient for transferring knowledge from the language domain to the ZSAR classification. Table 5 also suggests that the primary limitation on HMDB51 is related to the video sentence because there are no significant variations in accuracy taking different class label representations as there are on UCF101.

Table 5: Performance on the HMDB51 and UCF101 datasets considering separated sentences or paragraphs. All experiments were carried out on the TruZe protocol.

	HMDB51	UCF101
Baseline (only label)	19.5	36.6
Paragraph	19.5	43.2
Sentences	20.4	49.1

4.5.4. How is the performance affected if we change the language encoder?

Our method uses language encoders in two steps. In the first one, the encoder estimates the similarity between sentences from Internet documents and class labels, producing a semantic sentence space. In the second step, the encoder embeds sentences from semantic space and video observers to generate a joint embedding space. We can employ different language encoders in these two steps, as shown in Table 6. More specifically, we employ the Sentence2Vec [35] model and two paraphrase models from the *Sentence Transformers* repository: paraphrase-MiniLM-L6-v2 and paraphrase-distilroberta-base-v2. No models are fine-tuned or pretrained with our data. The results clearly show that encoding the joint embedding space with Sentence2Vec is unsuitable since this model cannot overcome the gap

between videos and class label descriptions, resulting in an accuracy close to the random value.

On the other hand, the adoption of pre-trained paraphrase-based models results in a strong performance because the model is optimized to learn similarities in sentence pairs. Using Sentence2Vec to preprocess the semantic information does not degrade the model performance at all. In this case, it is important to highlight that the comparison is made between the class label (which is not a sentence) and sentences. Therefore, this model can select sentences containing the exact label or synonyms. The performance combining Sentence2Vec with any paraphrase-based is lower than other configurations, possibly because the video descriptions are not enforced to present words contained in the class label in their sentences.

Table 6: Investigation on the semantic embedder for semantic preprocessing and ZSAR embedding. All experiments were performed on the TruZe protocol. Sent2Vec = Sentence2Vec, MiniLM = paraphrase-MiniLM-L6-v2, DR = paraphrase-distilroberta-base-v2.

Sem. Inf. Pre-proc.			ZSAR embedder			HMDB51	UCF101
Sent2Vec	MiniLM	DR	Sent2Vec	MiniLM	DR		
✓			✓			4.8	2.6
✓				✓		18.3	40.7
✓					✓	16.0	40.4
	✓		✓			7.5	1.5
	✓			✓		19.9	45.9
	✓				✓	19.9	48.2
		✓	✓			5.0	1.3
		✓		✓		20.5	46.3
		✓			✓	20.4	49.1

The observations in this experiment conduct us to the next question.

4.5.5. *What is the relation between the sentences quality and the ZSAR performance?*

We investigate this question by taking the model from *Observer 1* to compute the quality captioning measures (Meteor, Bleu@3, and Bleu@4) and ZSAR accuracy for each training epoch on UCF101. Training was stopped after ten epochs without improvements on Meteor. As expected, there is a high correlation ($r > 0.8$) between these measures, especially on Meteor ($r > 0.9$), as shown in Figure 6. Considering that video captioning is an active research topic with much room for improvement, the results suggest that better models for this task will directly imply higher accuracy.

4.5.6. *How is the performance with relaxed ZSAR constraints?*

ZSAR has extensive literature with several strategies for performing video embedding and class embedding, as detailed in [11]. Comparing these methods is not straightforward because several details on split configuration, random runs, and ZSAR constraints must be taken into account. As mentioned earlier, several deep learning-based video embeddings violate the ZSAR assumption by using 50% of the classes for testing. Our method is one of them and, due to this problem, we evaluate it under the TruZe protocol (Table 3). Nevertheless, a comparison under 50%/50% or 0%/50% protocols clarifies how good our method is compared to the broad literature. Additionally, analyzing the results reported in [16], we can assume that i3D pre-trained on the Kinetics400 dataset produces an overestimated performance of approximately 15%. Unfortunately, we cannot quantify the underestimation performance due to disregarding the training split since HMDB51 and UCF101 have no sentence annotations.

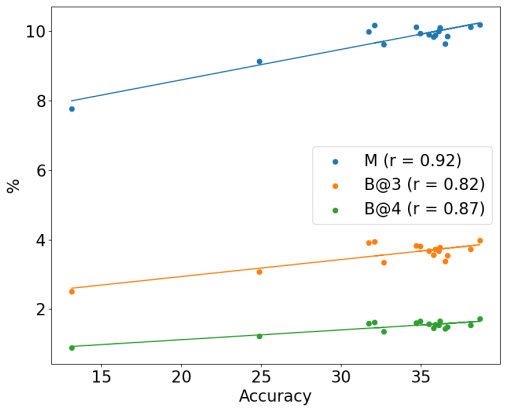


Figure 6: Comparison of captioning scores (Meteor, Bleu 3, and Bleu 4) and ZSAR accuracy under the TruZe protocol for Observer 1 at different training stages.

Table 7 is divided into two sections. The first groups the methods evaluated in the 50%/50% protocol, while the second groups the methods evaluated in the 0%/50% protocol (i.e., *cross-dataset*). In the latter, we immediately observe that the performance of our method on HMDB51 is much better than that of O2A. It is worth mentioning that this dataset was not used in the evaluation of other methods in this group, possibly because it is challenging to overcome the semantic gap due to short videos and generic actions. As an example, ER-ZSL [8] leverages object semantics in this dataset, but it improves generalization by concatenating visual features, which seems imperative to achieve higher performances.

Regarding the performance on UCF101, our method is on par with ER-ZSL, DASZL and CLASTER, which

Table 7: SOTA comparison under 50% / 50% and 0% / 50% splits reporting Top-1 accuracy (%) ± standard deviation. Our results were computed with 50 random runs. FV = fisher vector; BoW = bag of words; Obj = objects; S = image spatial feature; A = attribute; W_N = word embedding of class names, W_T = word embedding of class texts, ED = elaborative description; Sent = sentences.

Method	Video	Class	HMDB51	UCF101
50% / 50%				
DAP [25]	FV	A	N/A	15.9±1.2
IAP [25]	FV	A	N/A	16.7±1.1
HAA [28]	FV	A	N/A	14.9±0.8
SVE [55]	BoW	W_N	13.0±2.7	10.9±1.5
ESZSL [43]	FV	W_N	18.5±2.0	15.0±1.3
SJE [1]	FV	W_N	13.3±2.4	9.9±1.4
SJE [1]	FV	A	N/A	12.0±1.2
MTE [56]	FV	W_N	19.7±1.6	15.8±1.3
ZSECOC [39]	FV	W_N	22.6±1.2	15.1±1.7
UR [58]	FV	W_N	24.4±1.6	17.5±1.6
ASR [51]	C3D	W_T	21.8±0.9	24.4±1.0
LMR [38]	i3D	W_N	34.7±2.4	33.4±1.8
OutDist [30]	i3D+C3D	A	N/A	38.3±3.0
OutDist [30]	i3D+C3D	W_N	30.2±2.7	26.9±2.8
TS-GCN [12]	Obj	W_N	23.2±3.0	34.2±3.1
SFGAN [27]	i3D	W_N	32.4±4.1	29.8±2.8
E2E [3]	r(2+1)d	W_N	32.7	48
GAN-KG [46]	i3D	W_N	31.2±1.7	28.3±1.8
DASZL [22]	TSM	A	N/A	48.9±5.8
ER-ZSL [8]	(S+Obj)	ED	35.3±4.6	51.8±2.9
CLASTER [15]	i3D	W_N	41.8±2.1	50.2±3.8
0% / 50%				
O2A [21]	Obj	W_N	15.6	30.3
SAOE [31]	Obj	W_N	N/A	40.4±1.0
OP [32]	Obj	W_N	N/A	47.3
DO-SC [4]	Obj	S_{embs}	N/A	45.2±4.6
Ours	Sent	Sent	28.3±3.0	49.0±3.5

is impressive considering it is based entirely on transfer learning. Finally, comparing our approach with methods that also use i3D for visual embedding, the proposed method is on par with CLASTER and outperforms GAN-KG, SFGAN, LMR, and OutDist by a large margin, showing that its high performance is not only due to the bias from using i3D.

5. Conclusions and Future Work

In this work, we proposed to perform ZSAR by representing videos and semantic information with a common type of data: sentences in natural language. We trained two video captioning architectures with different input modalities in the ActivityNet Captions dataset and used these models to produce sentences for the HMDB51 and UCF101 videos. We then evaluated the ZSAR performance in a cross-dataset scenario.

Our conclusions are: (i) the textual descriptions provided by Observers are sufficient to outperform the state of the art in UCF101 and achieve a remarkable performance on HMDB51 (where clips have, on average, half time duration than UCF101); (ii) it is possible to perform ZSAR with pre-trained paraphrase models, leveraging the high availability of annotated data; no additional training or domain adaptation techniques were needed; (iii) we showed that the main performance limitation is the current state of the art on video captioning. However, the method is "plug and play" and enables us to replace the models with more accurate ones when they become available; and (iv) we chose to work only with captioning models, but models for other tasks can be used to provide semantic information, for example, object detection with replacing by concepts (as in [8]) or video tagging. We intend to investigate these possibilities in future work.

Acknowledgments

This work was supported by the Federal Institute of Paraná, Federal University of Paraná and by grants from the National Council for Scientific and Technological Development (CNPq) (grant numbers 309330/2018-1 and 308879/2020-1). The Titan Xp and Quadro RTX 8000 GPUs used for this research were donated by the NVIDIA Corporation.

References

[1] Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B., 2015. Evaluation of output embeddings for fine-grained image classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2927–2936.

[2] Banerjee, S., Lavie, A., 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72.

[3] Brattoli, B., Tighe, J., Zhdanov, F., Perona, P., Chalupka, K., 2020. Rethinking zero-shot video classification: End-to-end training for realistic applications, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4613–4623.

[4] Bretti, C., Mettes, P., 2021. Zero-shot action recognition from diverse object-scene compositions, in: British Machine Vision Conference (BMVC), pp. 1–14.

[5] Carreira, J., Noland, E., Hillier, C., Zisserman, A., 2019. A short note on the Kinetics-700 human action dataset. arXiv preprint arXiv:1907.06987.

[6] Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? a new model and the kinetics dataset, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4724–4733.

[7] Changpinyo, S., Chao, W.L., Gong, B., Sha, F., 2016. Synthesized classifiers for zero-shot learning, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5327–5336.

[8] Chen, S., Huang, D., 2021. Elaborative rehearsal for zero-shot action recognition, in: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 13638–13647.

[9] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 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), Association for Computational Linguistics, Minneapolis, Minnesota. pp. 4171–4186.

[10] Estevam, V., Laroca, R., Pedrini, H., Menotti, D., 2021a. Dense video captioning using unsupervised semantic information. arXiv preprint arXiv:2112.08455, 1–12.

[11] Estevam, V., Pedrini, H., Menotti, D., 2021b. Zero-shot action recognition in videos: A survey. Neurocomputing 439, 159–175.

[12] Gao, J., Zhang, T., Xu, C., 2019. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs, in: AAAI Conference on Artificial Intelligence, pp. 8303–8311.

[13] Gemmeke, J.F., Ellis, D.P.W., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M., 2017. Audio set: An ontology and human-labeled dataset for audio events, in: International Conference on Acoustics, Speech, & Signal Processing, pp. 776–780.

[14] Ghosh, P., Saini, N., Davis, L.S., Shrivastava, A., 2020. All about knowledge graphs for actions. arXiv preprint arXiv:2008.12432, 1–14.

[15] Gowda, S.N., Sevilla-Lara, L., Keller, F., Rohrbach, M., 2021a. CLASTER: clustering with reinforcement learning for zero-shot action recognition. arXiv preprint arXiv:2101.07042, 1–13.

[16] Gowda, S.N., Sevilla-Lara, L., Kim, K., Keller, F., Rohrbach, M., 2021b. A new split for evaluating true zero-shot action recognition, in: DAGM German Conference on Pattern Recognition (GCPR), pp. 1–15.

[17] Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., Saenko, K., 2013. YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition, in: IEEE International Conference on Computer Vision, pp. 2712–2719.

[18] Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C., 2015. ActivityNet: A large-scale video benchmark for human activity understanding, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 961–970.

[19] Hershey, S., Chaudhuri, S., Ellis, D.P.W., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., Slaney, M., Weiss, R.J., Wilson, K., 2017. CNN architectures for large-scale audio classification, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 131–135.

[20] Iashin, V., Rahtu, E., 2020. A better use of audio-visual cues: Dense video captioning with bi-modal transformer, in: British Machine Vision Conference (BMVC), pp. 1–16.

[21] Jain, M., van Gemert, J.C., Mensink, T., Snoek, C.G.M., 2015. Objects2Action: Classifying and localizing actions without any video example, in: IEEE International Conference on Computer Vision (ICCV), pp. 4588–4596.

[22] Kim, T.S., Jones, J.D., Peven, M., Xiao, Z., Bai, J., Zhang, Y., Qiu, W., Yuille, A., Hager, G.D., 2021. DASZL: Dynamic action signatures for zero-shot learning, in: AAAI Conference on Artificial Intelligence, pp. 1–10.

[23] Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C., 2017. Dense-captioning events in videos, in: International Conference on Computer Vision (ICCV), pp. 706–715.

[24] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T., 2011. HMDB: A large video database for human motion recognition, in: International Conf. on Computer Vision, pp. 2556–2563.

[25] Lampert, C.H., Nickisch, H., Harmeling, S., 2009. Learning to detect unseen object classes by between-class attribute transfer, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 951–958.

[26] Lashin, V., Rahtu, E., 2020. Multi-modal dense video captioning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 4117–4126.

[27] Lee, J., Kim, H., Byun, H., 2021. Sequence feature generation with temporal unrolling network for zero-shot action recognition. Neurocomputing 448, 313–323.

[28] Liu, J., Kuipers, B., Savarese, S., 2011. Recognizing human actions by attributes, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3337–3344.

[29] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 .

[30] Mandal, D., Narayan, S., Dwivedi, S.K., Gupta, V., Ahmed, S., Khan, F.S., Shao, L., 2019. Out-of-distribution detection for generalized zero-shot action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 9985–9993.

[31] Mettes, P., Snoek, C.G.M., 2017. Spatial-aware object embeddings for zero-shot localization and classification of actions, in: IEEE International Conference on Computer Vision, pp. 1–10.

[32] Mettes, P., Thong, W., Snoek, C., 2021. Object priors for classifying and localizing unseen actions. International Journal of Computer Vision 129, 1954–1971.

[33] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013. Distributed representations of words and phrases and their compositionality, in: International Conference on Neural Information Processing Systems (NeurIPS), pp. 3111–3119.

[34] Mohamed, M.A., Mertsching, B., 2012. TV-L1 optical flow estimation with image details recovering based on modified census transform, in: International Symposium on Visual Computing (ISVC), pp. 482–491.

[35] Pagliardini, M., Gupta, P., Jaggi, M., 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features, in: Conference of the North American Chapter of the Association for Computational Linguistics, pp. 528–540.

[36] Papineni, K., Roukos, S., Ward, T., Zhu, W., 2002. BLEU: a method for automatic evaluation of machine translation, in: Annual Meeting of the Association for Computational Linguistics (ACL), pp. 311–318.

[37] Pennington, J., Socher, R., Manning, C.D., 2014. GloVe: Global vectors for word representation, in: Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543.

[38] Piergiovanni, A., Ryoo, M.S., 2018. Learning shared multimodal embeddings with unpaired data. CoRR abs/1806.08251.

[39] Qin, J., Liu, L., Shao, L., Shen, F., Ni, B., Chen, J., Wang, Y., 2017. Zero-shot action recognition with error-correcting output codes, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1042–1051.

[40] Reimers, N., Gurevych, I., 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. CoRR abs/1908.10084.

[41] Reimers, N., Gurevych, I., 2020. Making monolingual sentence embeddings multilingual using knowledge distillation, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4512–4525.

[42] Roitberg, A., Martinez, M., Haurilet, M., Stiefelhagen, R., 2018. Towards a fair evaluation of zero-shot action recognition using external data, in: European Conference on Computer Vision (ECCV) Workshops, pp. 1–9.

[43] Romera-Paredes, B., Torr, P.H.S., 2015. An embarrassingly sim-

ple approach to zero-shot learning, in: International Conference on Machine Learning, p. 2152–2161.
[44] Soomro, K., Zamir, A.R., Shah, M., 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 1–6.
[45] Speer, R., Chin, J., Havasi, C., 2017. ConceptNet 5.5: An open multilingual graph of general knowledge, in: AAAI Conference on Artificial Intelligence, pp. 4444–4451.
[46] Sun, B., Kong, D., Wang, S., Li, J., Yin, B., Luo, X., 2021. GAN for vision, KG for relation: a two-stage deep network for zero-shot action recognition. arXiv preprint arXiv:2105.11789.
[47] Sun, D., Yang, X., Liu, M., Kautz, J., 2017. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. arXiv preprint arXiv:1709.02371, 1–18.
[48] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M., 2015. Learning spatiotemporal features with 3D convolutional networks, in: IEEE International Conference on Computer Vision (ICCV), pp. 4489–4497.
[49] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: International Conference on Neural Information Processing, pp. 6000–6010.
[50] Wang, H., Schmid, C., 2013. Action recognition with improved trajectories, in: IEEE International Conference on Computer Vision (ICCV), pp. 3551–3558.
[51] Wang, Q., Chen, K., 2017a. Alternative semantic representations for zero-shot human action recognition, in: Machine Learning and Knowledge Discovery in Databases, pp. 87–102.
[52] Wang, Q., Chen, K., 2017b. Zero-shot visual recognition via bidirectional latent embedding. International Journal of Computer Vision 124, 356–383.
[53] Wu, Z., Fu, Y., Jiang, Y., Sigal, L., 2016. Harnessing object and scene semantics for large-scale video understanding, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3112–3121.
[54] Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B., 2016. Latent embeddings for zero-shot classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 69–77.
[55] Xu, X., Hospedales, T., Gong, S., 2015. Semantic embedding space for zero-shot action recognition, in: IEEE International Conference on Image Processing (ICIP), pp. 63–67.
[56] Xu, X., Hospedales, T., Gong, S., 2016. Multi-task zero-shot action recognition with prioritised data augmentation, in: European Conference on Computer Vision (ECCV), pp. 343–359.
[57] Zhang, B., Hu, H., Sha, F., 2018. Cross-modal and hierarchical modeling of video and text, in: European Conference on Computer Vision (ECCV), pp. 385–401.
[58] Zhu, Y., Long, Y., Guan, Y., Newsam, S.D., Shao, L., 2018. Towards universal representation for unseen action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9436–9445.