Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-16566-5

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

Valter Estevam[1,2] · Rayson Laroca[2] · Helio Pedrini[3] · David Menotti[2]

Received: 15 February 2022 / Revised: 18 July 2023 / Accepted: 18 August 2023

Abstract
This paper presents a novel approach to Zero-Shot Action Recognition. Recent works have explored the detection and classification of objects to obtain semantic information from videos with remarkable performance. Inspired by them, we propose using video captioning methods to extract semantic information about objects, scenes, humans, and their relationships. To the best of our knowledge, this is the first work to represent both videos and labels with descriptive sentences. More specifically, we represent videos using sentences generated via video captioning methods and classes using sentences extracted from documents acquired through search engines on the Internet. Using these representations, we build a shared semantic space employing BERT-based embedders pre-trained in the paraphrasing task on multiple text datasets. The projection of both visual and semantic information onto this space is straightforward, as they are sentences, enabling classification using the nearest neighbor rule. We demonstrate that representing videos and labels with sentences alleviates the domain adaptation problem. Additionally, we show that word vectors are unsuitable for building the semantic embedding space of our descriptions. Our method outperforms the state-of-the-art performance on the UCF101 dataset by 3.3 p.p. in accuracy under the TruZe protocol and achieves competitive results on both the UCF101 and HMDB51 datasets under the conventional protocol (0/50% - training/testing split). Our code is available at https://github.com/valterlej/zsarcap.

Keywords Cross-dataset learning · Paraphrase estimation · Video captioning · Zero-shot learning

1 Introduction

Human Action Recognition (HAR) is an active research topic in computer vision. Several supervised models have been proposed with impressive performance in the last years, especially those based on deep learning [1]. At the same time, large-scale datasets contain-

✉ Valter Estevam
valter.junior@ifpr.edu.br

Extended author information available on the last page of the article

ing a massive number of human actions, such as Kinetics-400 [2], Kinetics-700 [3] and ActivityNet [4], 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 [5].

Zero-Shot Learning (ZSL) [6, 7] and their applications to actions, Zero-Shot Action Recognition (ZSAR) [5, 8, 9], are computer vision tasks that emerge 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, ZSAR models need to represent the class labels with semantic information, and the classification is performed with some function, usually learned with known classes by correlating visual patterns with the label semantic properties [10].

Traditionally, the videos are represented using spatio-temporal features (e.g., Improved Dense Trajectories (IDT) [11], Convolutional 3D Network (C3D) [12] or Inflated 3D Network (I3D) [2]), and the class labels are represented with attributes or word vectors such as Word2Vec [13] or Global Vectors (GloVE) [14].

Although this general scheme (deep features $\leftrightarrow$ word vectors) 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 [5]. 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 [10].

As representing classes with a set of attributes is not scalable, some recent approaches have replaced attributes by detecting objects in scenes [15, 16]. This approach works because the visual class-object relationships also exist in texts and are captured in word vectors [16]. 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, aiming to mitigate the domain adaptation problem. 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 Fig. 1. Although intuitive, this is the first work, to the best of our knowledge, that uses neural networks to convert videos into descriptive sentences, and then to perform Zero-Shot Action Recognition (ZSAR) with these sentences.

First, we encode the videos using observers that generate a descriptive sentence given an input video, as shown in Fig. 1(a). We choose SOTA video captioning architectures from [17–19] and pre-training them in the ActivityNet captions dataset (i.e., without any class label). These architectures present remarkable properties, such as (i) using self-attention to concentrate on more relevant 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.

We then encode the action labels with texts collected from the Internet through search engines, as illustrated in Fig. 1(b). More specifically, we use the descriptions provided by Wang and Chen [20] and employ a simple strategy to select only the sentences most closely

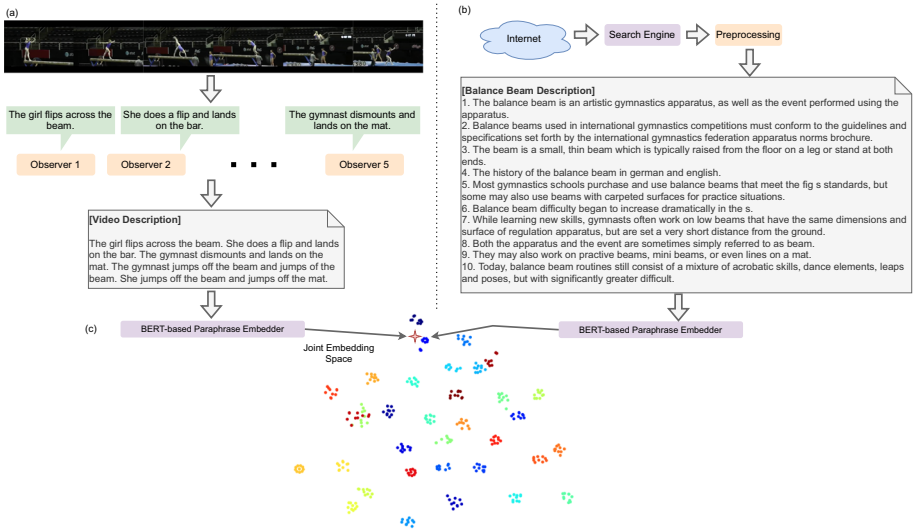

Fig. 1 Representation of our ZSAR method. In (a), we show the visual representation procedure. In (b), the semantic representation is shown. Finally, in (c), the joint embedding

related to the action labels. We demonstrate this procedure is more effective then those proposed by Chen and Huang [8] and our final class description is independent of human evaluation or approval.

As shown in Fig. 1(c), we take advantage of SOTA paraphrase methods based on Bidirectional Encoder Representations from Transformers (BERT), and produce a joint embedding space in which a simple Nearest Neighbour (NN) method achieves remarkable performance.

Our work has some advantages compared to existing methods: (1) 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; (2) a joint latent representation between visual patterns and texts is encoded in video captioning neural networks, being a natural bridge between these information modalities; (3) 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 (4) 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:

1. 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;
2. 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 human evaluation of their content;
3. 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;

4. 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 by creating new end-to-end models combining these two objectives (captioning and ZSAR).

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 Estevam et al. [10], existing methods based on attributes manually annotated 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. [21] 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. [15] 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.

Zuxuan et al. [22] 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 [16], 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. [23] 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 [24], where the concepts are represented with word vectors. The second stream used the visual representations of objects (with the methods used in [15] and [16]) 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).

Ghosh et al. [25] were inspired by [23]. In their work, knowledge graphs were fed 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. [26] 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 [2].

2.2 Text representations for ZSAR

Zhang et al. [27] proposed an improved model for learning visual and textual alignments. Typically, 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 [28] 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 NN 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: (1) 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; (2) our method does not require any class label annotation nor to train the ZSAR clas-

sifier; (3) 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 (4) 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 Fig. 1. To facilitate our presentation, Table 1 summarizes the notations used in this paper.

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

Table 1 Nomenclature used in our work

Notation	Description
$\mathcal{Y}_s, \mathcal{Y}_u$	sets of labels for labeled and unlabeled action classes
y_{u_s}, y_{u_n}	seen and unseen action classes
$y_{\text{pred}}, y_{\text{prot}}$	predicted class label, prototype of an action class (a descriptive sentence)
$\mathcal{Y}u_{\text{prots}}$	set of label representations (textual description) for unlabeled action classes
$\emptyset$	empty set
v	a video (visual and audio streams)
n_c	a stack of video features for a video
v_f	a set of n_c visual features
Y	a set of m words of a sentence encoded with BMT or Transformer
d_{model}	dimension of the internal encoding layer of BMT or Transformer
Q, K, V	queries, keys and values (inputs of a self-attention layer)
d_k	dimension of the queries and keys
$W, W_1, W2$	internal weight matrices
V_f, A, Sm	a visual stream, an audio stream, a semantic stream (VisGloVe [19])
ASm	an audio or a semantic stream depending on the Observer input
pos	the position of a feature or word in a BMT or Transformer input
i	number of column indices used on positional encoding
PE	indicates that the feature was yielded by a positional encoding layer
self	indicates that the feature was yielded by a self-attention layer
-att	indicates that the feature was yielded by a multi-head attention layer
FFN	indicates that the features were yielded by a feed-forward network
u_a, u_b	sentences a and b feed to the Siamese BERT networks [29]
n_s	dimension of sentence embeddings
k	number of labels in the paraphrasing classification pre-training
s_a, s_p, s_n	sentence embeddings for anchors, positive, and negative sentences (Sentence-BERT training)

is respected:

$$\mathcal{Y}_u \cap \mathcal{Y}_s = \emptyset \tag{1}$$

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 NN rule under some similarity function, such as

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

in which $Sim(\cdot)$ 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 [17] (using the original transformer implementation from [30]), and Bi-Modal Transformer (BMT) [18]. 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.

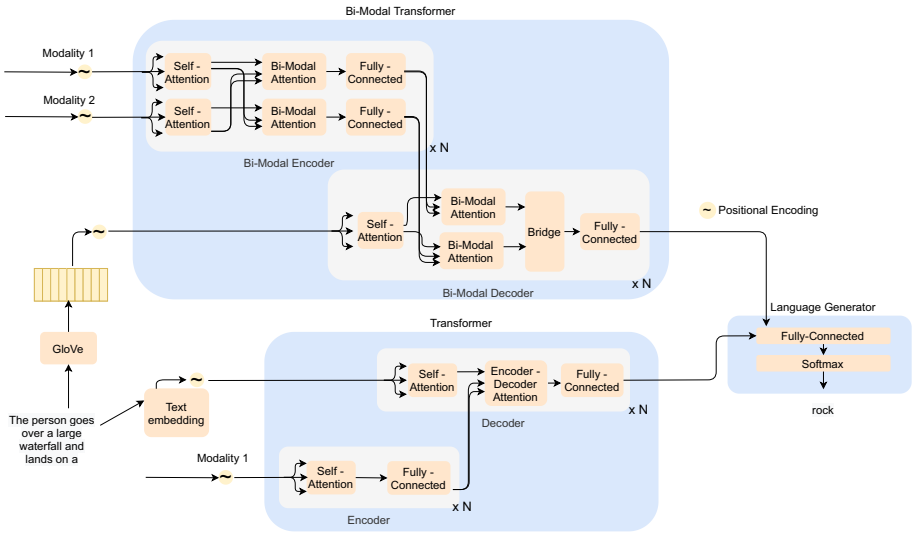

Fig. 2 Overview of the captioning architectures showing the BMT and Transformer layers with their inputs and the language generation module. Adapted from [10]

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

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

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 (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

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

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.

Subsequently, a multi-head attention layer processes these representations with scaled dot-product attention defined in terms of queries (Q), keys (K), and values (V) as

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

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

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

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

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

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

$$FFN(u) = \max(0, u \times W_1 + b_1) \times W_2 + b_2, \tag{8}$$

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 (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

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

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 Bi-Modal Transformer (BMT). Considering the encoder, this transformer has two differences from the Transformer encoder. It takes two streams, visual V_f and audio A [18] or semantic Sm [19], separately. We denote this second stream as ASm (i.e., audio or semantic). The encoder has three sub-layers: self-attention (5), producing $V_f^{self-att}$ and $ASm^{self-att}$; bi-modal attention, i.e.,

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

and

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

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 (6). Afterward, the bi-modal attention is computed as

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

and

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

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

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

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 [19] 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 $\mathcal{Y}_{\text{prot}} = \{y_{\text{prot}_1}, y_{\text{prot}_2}, ..., y_{\text{prot}_q}\}$ 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) [29] 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 [29]. It is an improved BERT [31] 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 Fig. 3.

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

$$o = softmax(W_t(u_a, u_b, |u_a - u_b|))) \tag{15}$$

where $|u_a - u_b|$ is an element-wise subtraction, $W_t \in \mathbb{R}^{3n_s \times k}$ is the trainable weights, n_s 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_a and u_b is computed, and the loss function is the MSE.

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

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

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.

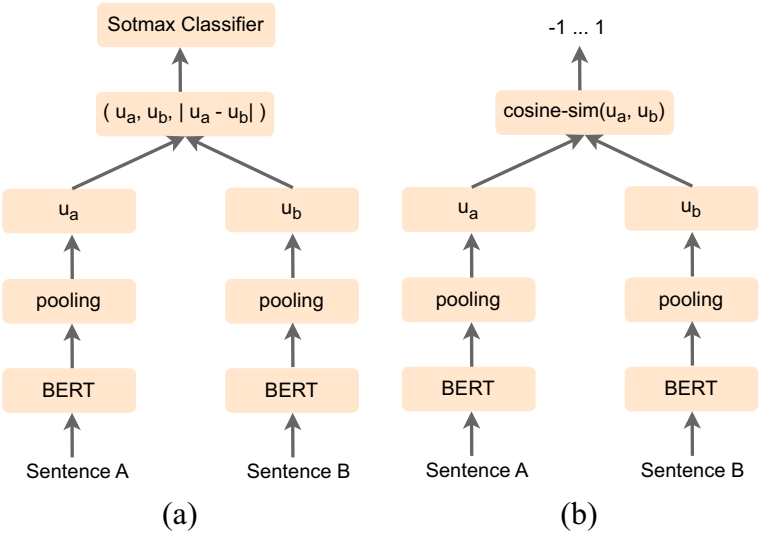

Fig. 3 SBERT architecture from Reimers and Gurevych [29]. In (a) is shown the classification objective function, and in (b) the architecture used at the inference or regression tasks

Our interest is in the vector u (see Fig. 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 [31] and [32], 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 protocols

Our observers were trained using the ActivityNet Captions dataset [33], 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 [34] and UCF101 [35]. The former is composed of 6,766 videos from 51 classes, illustrated in Fig. 4, 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, illustrated in Fig. 5, with frame resolution standardized to 25 FPS and 320 × 240 pixels. The average

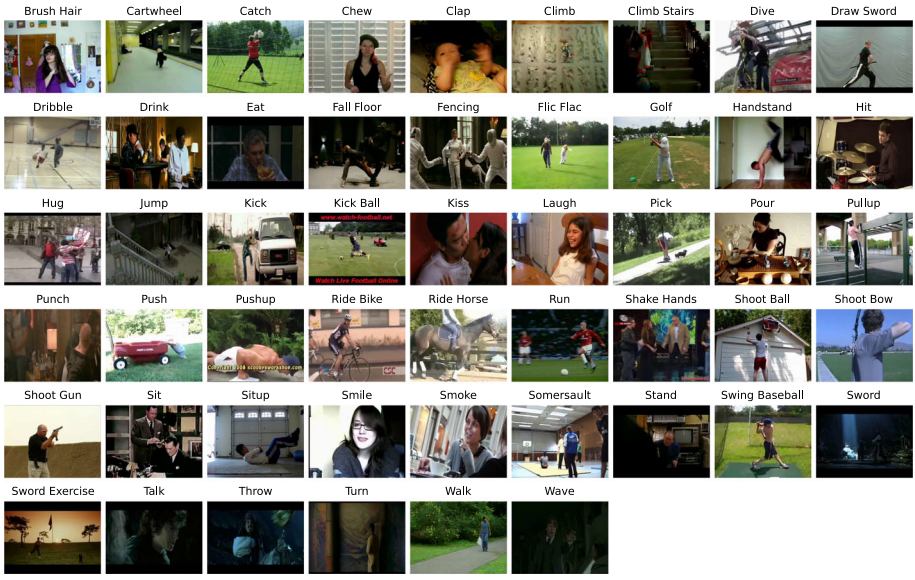

Fig. 4 Samples for the 51 action classes from the HMDB51 dataset [34]

Fig. 5 Samples for the 101 action classes from the UCF101 dataset [35]

duration of the videos is 7.2s. As is customary in ZSAR research [10], performance is evaluated using the well-known accuracy metric, which quantifies the number of correct predictions relative to the total number of predictions made.

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 [2], 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 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 [8, 10, 36–38].

Taking this into account, we adopt the TruZe evaluation protocol [38] on UCF101 and HMDB51 datasets, 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 configuration 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)[1].

Finally, we also provide a comparison using a conventional protocol employed in most of the works. In some cases 0/50%, and in the most 50%/50%[2]. Although there are overlaps in training and testing sets, several methods employ this scheme [20, 39–41]. This evaluation is important to observe the impact of the use of I3D features on the results and how our method compares to others independently of the adopted protocol.

4.2 Implementation details

We compute features as shown in Fig. 6. 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 [42] optical flow algorithm for the PWC-Net [43], as it is much faster[3]. 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 [44] pre-trained on AudioSet [45]. 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 [19] 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 Fig. 6(a)). With these features, we fed two architectures for video captioning (i.e., Transformer and Bi-Modal Transformer (BMT)) which allowed us to generate 5 distinct observers. Fig. 6(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 [46] 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 [47] to allow evaluating the quality of the sentences produced during the training stage. The Visual

[1] **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.

[2] Not all methods allow 0/50 experiments.

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

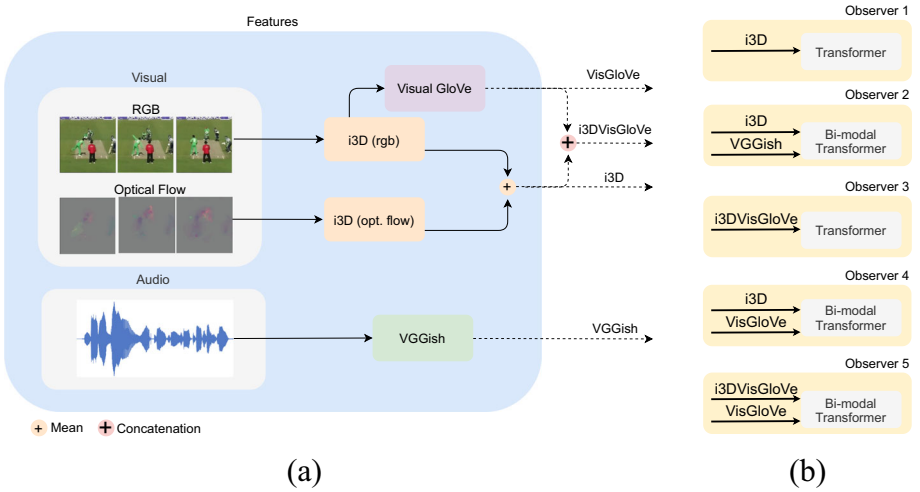

Fig. 6 Features and observers. In (a) is shown features computed from visual and audio streams, and in (b) the observers architecture and their respective input features

Global Vectors (VisGloVE) 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 would produce different sentences given a sample video. Although different, these sentences would 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 use the textual descriptions provided in [20][4] as side information. The texts are processed using the NLTK[5] package for splitting paragraphs into sentences and the *contractions*[6] 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 (cosine similarity) compared to the label encoding. The sentences from the observers are concatenated. We build the joint space with Sentence-BERT encoders [29], namely, the *paraphrase-distilroberta-base-v2*[7] model [48]. A NN algorithm employing cosine distance is used to conduct the ZSAR classification.

The deep learning models were implemented using PyTorch[8], while the ZSAR classifier was implemented using scikit-learn[9]. All experiments were conducted on a computer system

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

[5] https://www.nltk.org/

[6] https://pypi.org/project/contractions/

[7] 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.

[8] https://pytorch.org/

[9] https://scikit-learn.org/

equipped with an AMD Ryzen 7 2700X 3.7GHz CPU, 64 GB of RAM, and an NVIDIA Titan Xp GPU (12 GB). The experiments were executed on the Ubuntu operating system.

4.3 Selected benchmarks and evaluation

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

Latem [49] 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 [50] generates a weighted graph with synthesized classes that ensure the alignment between semantic embedding space and the classifier space by minimizing the distortion error. BiDiLEL [5] learns two projection functions for projecting visual and semantic spaces onto a shared embedding space to preserve the relationship between them.

OutDist [39] learns a visual feature synthesizer given the semantics and an out-of-distribution detector to distinguish generated features from seen ones. WGAN [51] is a model that synthesizes CNN features conditioned on class-level semantic information. It provides a way to generate a class-conditional feature distribution conditioned by a semantic descriptor. E2E [36] 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, UCF101 and HMDB51. Finally, CLASTER [52] applies reinforcement learning on the clustering of visual-semantic embeddings[10].

4.4 Results

Table 2 shows a comparison with the selected baselines. As can be seen, the proposed method achieves state-of-the-art performance on the UCF101 dataset, even without using the 67 classes from the training set. The HMDB51 dataset is challenging due to their actions (e.g., run, turn, punch, chew, clap) are complex to define through text and due to their short video clips that do not take advantage of the Transformer architecture benefits. Despite these issues, we obtain a remarkable performance.

ZSAR has an extensive literature, with several strategies for performing video embedding and class embedding, as detailed in [10]. Comparing these methods is not straightforward because several details on split configuration, random runs, and ZSAR constraints must be taken into account. As mentioned previously, several deep learning-based video embeddings violate the ZSAR assumption when using 50% of the classes for testing. Considering that several works fail in preserving this premise [28, 39–41], a comparison under 50%/50% or 0%/50% protocols clarifies how good our method is compared to the broad literature.

Table 3 summarizes the performance on HMDB51 and UCF101 datasets for 28 different methods including ours. In this table, 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, and Sent = sentences are the strategy adopted to perform video embedding. When the model uses a different number of classes in training, we indicate this by including this number next to the accuracy value.

There are two sections in Table 3. The first groups the methods evaluated in the 50%/50% protocol, whereas the second groups the methods evaluated in the 0%/50% protocol (i.e., *cross-dataset*).

[10] A more detailed description for these methods can be found in [10]

Table 2 SOTA comparison under the TruZe protocol [38]. tr/te = train/test split configuration; Acc = accuracy

	HMDB51		UCF101	
	tr/te	Acc.	tr/te	Acc.
Latem [49] (CVPR'16)	29/22	9.4	67/34	15.9
SYNC [50] (CVPR'16)	29/22	11.6	67/34	15.0
BiDiLEL [5] (IJCV'17)	29/22	10.5	67/34	16.0
WGAN [51] (CVPR'18)	29/22	21.1	67/34	22.5
OutDist [39] (CVPR'19)	29/22	21.7	67/34	23.4
E2E [36] (CVPR'20)	29/22	31.5	67/34	45.2
CLASTER [52] (ECCV'22)	29/22	**33.2**	67/34	45.8
SPOT [53] (CVPRW'23)	29/22	24.0	67/34	25.5
Ours	0/22	20.4	0/34	**49.1**

To compare the results, we follow [10] and assume that the mean accuracy has a normal distribution and approximate the population standard deviation σ by sample standard deviation s. Therefore, the mean accuracy of population can be estimated by $\mu \approx \bar{x} \pm E$, where $E \approx t_{95\%,n-1}\frac{s}{\sqrt{n}}$ and $n-1$ are the degrees of freedom for n runs.

Considering this, we compare our results against the methods in which it is possible to estimate the mean accuracy with an error of 2% at 95% of confidence. Regarding the performance on UCF101, our method is on par with ER-ZSL, UR, SignleGAN, CLASTER (no statistical difference), which is impressive considering that it is based entirely on transfer learning. Methods such as E2E, PS-ZSAR or ViSET-96 are not directly comparable to our method since they do not provide the standard deviation value.

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, demonstrating that its high performance is not only due to the bias from using I3D. Unfortunately, we cannot quantify the underestimation performance due to disregarding the training split since HMDB51 and UCF101 datasets have no sentence annotations.

Considering the performance of our method on HMDB51 under 0/50%, it is superior to O2A. It is worth mentioning that this dataset was not used in the evaluation of other methods in this group, possibly because it is very challenging to overcome the semantic gap due the simple 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 as those obtained by CLASTER or SPOT.

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 2.

4.5.1 What is the impact of each observer or combination of observers on the performance?

In Table 4, 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

Table 3 SOTA comparison under 50% / 50% and 0% / 50% splits reporting Top-1 accuracy (%) ± standard deviation. Our results were computed with 50 random runs

Method	Video	Class	HMDB51	UCF101
50% / 50%				
DAP [54] (CVPR'09)	FV	A	N/A	15.9±1.2
IAP [54] (CVPR'09)	FV	A	N/A	16.7±1.1
HAA [55] (CVPR'11)	FV	A	N/A	14.9±0.8
SVE [56] (ICIP'15)	BoW	W_N	13.0±2.7	10.9±1.5
ESZSL [57] (ICML'15)	FV	W_N	18.5±2.0	15.0±1.3
SJE [58] (CVPR'15)	FV	W_N	13.3±2.4	9.9±1.4
SJE [58] (CVPR'15)	FV	A	N/A	12.0±1.2
MTE [59] (ECCV'16)	FV	W_N	19.7±1.6	15.8±1.3
ZSECOC [60] (CVPR'17)	FV	W_N	22.6±1.2	15.1±1.7
ASR [20] (ECML PKDD'17)	C3D	W_T	21.8±0.9	24.4±1.0
UR [61] (CVPR'18)	FV	W_N	N/A	42.5 ± 0.9 (200)
OutDist [39] (CVPR'19)	i3D+C3D	A	N/A	38.3±3.0
OutDist [39] (CVPR'19)	i3D+C3D	W_N	30.2±2.7	26.9±2.8
TS-GCN [23] (AAAI'19)	Obj	W_N	23.2±3.0	34.2±3.1
LMR [28] (WACV'20)	i3D	W_N	34.7±2.4	33.4±1.8
E2E [36] (CVPR'20)	r(2+1)d	W_N	32.7 (664)	48 (664)
SFGAN [40] (Neurocomputing'21)	i3D	W_N	32.4±4.1	29.8±2.8
DASZL [26] (AAAI'21)	TSM	A	N/A	48.9±5.8
ER-ZSL [8] (ICCV'21)	(S+Obj)	ED	35.3±4.6	51.8±2.9
PS-ZSAR [62] (NeurIPS 21)	r(2+1)d	W_T	33.8 (664)	49.2 (664)
GAN-KG [41] (PR'22)	i3D	W_N	31.2±1.7	28.3±1.8
Single-GAN [63] (VISAPP'22)	i3D	ResNet101	N/A	45.9±3.42
CLASTER [52] (ECCV'22)	i3D	W_N	**41.8±2.1**	50.2±3.8
SPOT [53] (CVPRW'23)	i3D+C3D + SPOT	W_N	39.8±1.4	42.8±1.7
ViSET-96 [64] (CVPRW'23)	ViSET	W_T	34.5 (564)	**53.2** (564)
0%/50%				
O2A [15] (ICCV'15)	Obj	W_N	15.6	30.3
SAOE [16] (ICCV'17)	Obj	W_N	N/A	40.4±1.0
OP [9] (IJCV'21)	Obj	W_N	N/A	47.3
DO-SC [65] (BMVC'21)	Obj	S_{embs}	N/A	45.2±4.6
Ours	Sent	Sent	**28.3±3.0**	**49.0±3.5**

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 Fig. 6(b)). As mentioned earlier, approximately 50% of the videos have audio signal. However, this observer has a

Table 4 Observer accuracy for the UCF101 and HMDB51 datasets under TruZe protocol. 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**

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 the impact of stack length, 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 5 shows the results acquired with these features taking the same pre-trained models used in Table 4. 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

Table 5 Observer accuracy for the HMDB51 dataset TruZe protocol changing the number of frames used to compute visual features from 24 to 10 and 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

Table 6 ZSAR performance on the HMDB51 and UCF101 datasets under TruZe protocol considering different semantic information modalities

	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**

features, which encode co-occurrence of visual patterns in complex events with long duration (one minute on average with a window of 24s) [19]. 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).

4.5.2 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 6 compares the ZSAR performance considering four scenarios: only class label, Elaborative Descriptions (ED), Ours + Elaborative Descriptions (ED), and only Ours.

The results in both datasets show that the proposed pre-processing method achieves a higher accuracy compared to others. Although Elaborative Descriptions (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.

Considering these results, we propose the following question:

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

Figures 7(a) and 7(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.

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 Fig. 7(a) illustrates another aspect of why HMDB51 is so challenging for our method. The configurations with 3 or 5 words and only one sentence

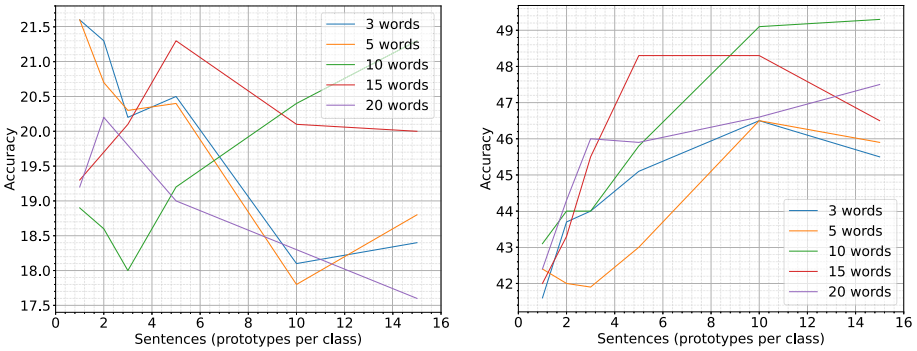

Fig. 7 ZSAR performance changing the maximum sentences per class and the minimum words per sentence in the prototypes. (a) results from HMDB51 and (b) from UCF101

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 6.

4.5.4 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 7 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 7 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.

4.5.5 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

Table 7 Performance on the HMDB51 and UCF101 datasets under TruZe protocol considering separated sentences or paragraphs

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

Table 8 Investigation on the semantic embedder for semantic pre-processing and Zero-Shot Action Recognition (ZSAR) embedding

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**

Experiments performed on the TruZe protocol

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 8. More specifically, we employ the Sentence2Vec [66] model and two paraphrase models from the *Sentence Transformers* repository: paraphrase-MiniLM-L6-v2 and paraphrase-distilroberta-base-v2. They are referred in Table 8 as Sent2Vec, MiniLM, and DR, respectively. No models are fine-tuned or pre-trained 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 pre-process 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.

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

4.5.6 What are the main limitations of our method?

In this subsection, we investigate two limiting aspects of our approach: the current SOTA in video captioning and the inter-class similarity. First, we examine the limitation of SOTA 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. The training was halted after ten epochs without improvements in Meteor. As expected, there is a strong correlation ($r > 0.8$) between these measures, especially on Meteor ($r > 0.9$), as shown in Fig. 8. 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 lead to higher accuracy.

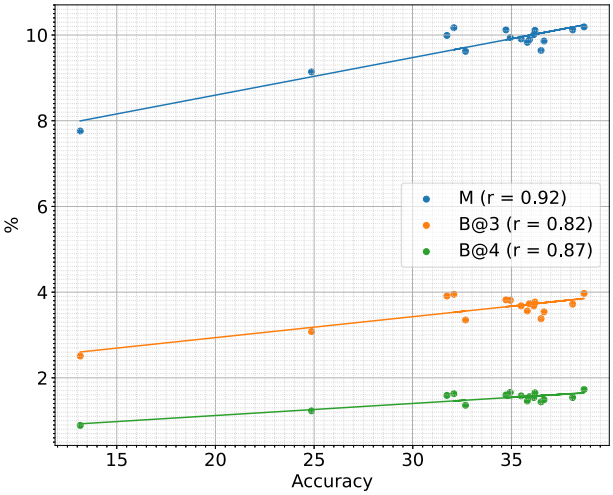

Fig. 8 Comparison of captioning scores (Meteor, Bleu 3, and Bleu 4) and ZSAR accuracy under the TruZe protocol for Observer 1 at different training stages

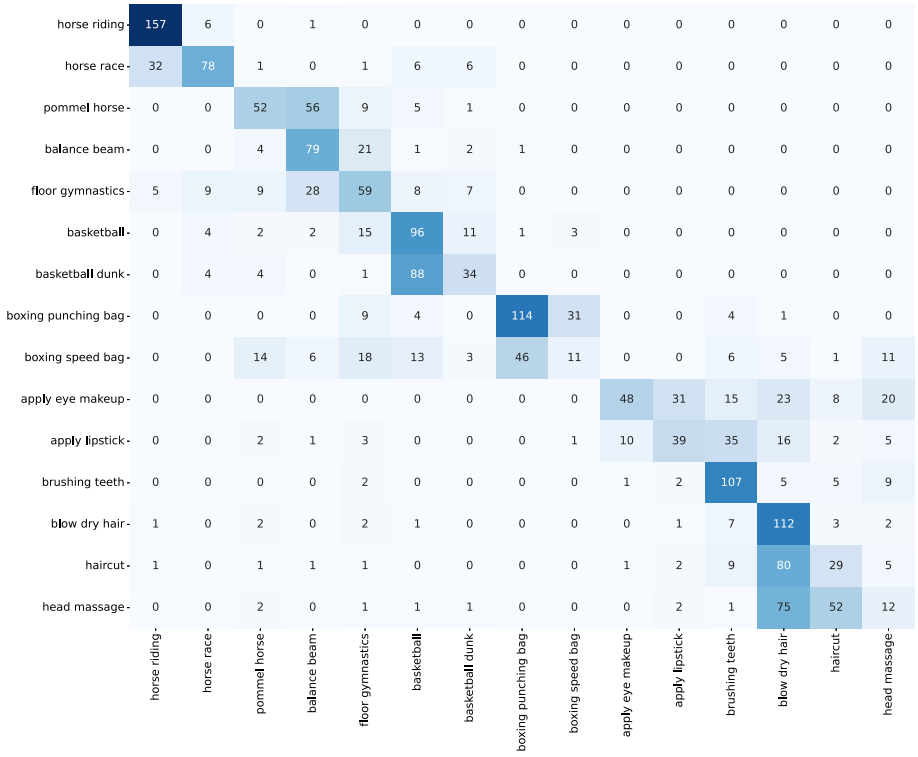

Fig. 9 Evaluation on the inter-class performance considering the complete method (5 observers) on UCF101

To conduct a more comprehensive investigation into inter-class performance, we selected a subset of 15 classes from UCF-101 that present challenging examples due to their high inter-class similarity. These classes can be divided into six groups: (1) activities involving horses, such as *horse riding* and *horse race*; (2) gymnastic performances, including *pommel horse*, *balance beam*, and *floor gymnastics*; (3) activities involving basketballs, such as *basketball* and *basketball dunk*; (4) boxing-related actions, namely *boxing punching bag* and *boxing speed bag*; (5) activities involving the face, such as *applying eye makeup*, *applying lipstick*, and *brushing teeth*; and (6) actions related to hair, such as *blow drying hair*, *getting a haircut*, and *receiving a head massage*.

Figure 9 clearly shows that the primary cause of errors lies in the high inter-class similarity (e.g., subgroups 4 – boxing-related, 5 – involving the face, 6 – related to hair). The results indicate the need to extract more discriminative features from individual frames or short clips, which can be accomplished by incorporating object relationships or other semantic features.

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:

1. The textual descriptions provided by Observers proved to be sufficient for outperforming state-of-the-art performance on UCF101 and achieving remarkable results on HMDB51, even considering the relatively shorter time duration of clips in HMDB51 compared to UCF101. Nevertheless, it is necessary to consider a combination of Observers to achieve better results;
2. ZSAR can be effectively conducted using pre-trained paraphrase models, capitalizing on the abundance of available data, without requiring any additional training or domain adaptation techniques;
3. We demonstrated a correlation between Meteor score and ZSAR accuracy, highlighting that the primary factor limiting performance is the current state of the art in video captioning. The proposed method is "plug and play", allowing for the seamless replacement of models with more accurate ones as they become available. Furthermore, future research can explore the integration of captioning and ZSAR into an end-to-end model, optimizing their shared objectives;
4. We specifically focused on working with captioning models in this study, but it is worth noting that models for various other tasks can also be employed to offer semantic information; for example, object detection with replacing by concepts (as in [8]) or video tagging. We acknowledge these possibilities and plan to investigate them in future research.

Acknowledgements 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 304836/2022-2 and 308879/2020-1). The Titan Xp GPU used for this research were donated by the NVIDIA Corporation.

Compliance with ethical standards

Competing of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Kong Y, Fu Y (2022) Human action recognition and prediction: A survey. Int J Comput Vis 130(5):1366–1401. https://doi.org/10.1007/s11263-022-01594-9
2. Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: Conf Comput Vis Pattern Recognit, pp 4724–4733. https://doi.org/10.1109/CVPR.2017.502
3. Carreira J, Noland E, Hillier C, Zisserman A (2019) A short note on the Kinetics-700 human action dataset. arXiv preprint arXiv:1907.06987
4. Heilbron FC, Escorcia V, Ghanem B, Niebles JC (2015) ActivityNet: A large-scale video benchmark for human activity understanding. In: Conf Comput Vis Pattern Recognit, pp 961–970. https://doi.org/10.1109/CVPR.2015.7298698
5. Wang Q, Chen K (2017) Zero-shot visual recognition via bidirectional latent embedding. Int J Comput Vis 124(3):356–383. https://doi.org/10.1007/s11263-017-1027-5
6. Xie Y, He X, Zhang J, Luo X (2020) Zero-shot recognition with latent visual attributes learning. Multimed Tools Appl 79:27321–27335. https://doi.org/10.1007/s11042-020-09316-4
7. Wang Y, Zhang H, Zhang Z, Long Y (2020) Asymmetric graph based zero shot learning. Multimed Tools Appl 79:33689–33710. https://doi.org/10.1007/s11042-019-7689-y
8. Chen S, Huang D (2021) Elaborative rehearsal for zero-shot action recognition. In: Int Conf Comput Vis, pp 13638–13647. https://doi.org/10.1109/ICCV48922.2021.01338
9. Mettes P, Thong W, Snoek CGM (2021) Object priors for classifying and localizing unseen actions. Int J Comput Vis 129:1954–1971. https://doi.org/10.1007/s11263-021-01454-y
10. Estevam V, Pedrini H, Menotti D (2021) Zero-shot action recognition in videos: A survey. Neurocomputing 439:159–175. https://doi.org/10.1016/j.neucom.2021.01.036
11. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: Int Conf Comput Vis, pp 3551–3558. https://doi.org/10.1109/ICCV.2013.441
12. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. In: Int Conf Comput Vis, pp 4489–4497. https://doi.org/10.1109/ICCV.2015.510
13. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. Conf Neural Inf Process Syst 2:3111–3119
14. Pennington J, Socher R, Manning CD (2014) GloVe: Global vectors for word representation. In: Conf Empir Methods Nat Lang Process, pp 1532–1543. https://doi.org/10.3115/v1/D14-1162
15. Jain M, van Gemert JC, Mensink T, Snoek CGM (2015) Objects2Action: Classifying and localizing actions without any video example. In: Int Conf Comput Vis, pp 4588–4596. https://doi.org/10.1109/ICCV.2015.521
16. Mettes P, Snoek CGM (2017) Spatial-aware object embeddings for zero-shot localization and classification of actions. In: Int Conf Comput Vis, pp 1–10. https://doi.org/10.1109/ICCV.2017.476
17. Iashin V, Rahtu E (2020) Multi-modal dense video captioning. In: Conf Comput Vis Pattern Recognit Workshops, pp 4117–4126. https://doi.org/10.1109/CVPRW50498.2020.00487
18. Iashin V, Rahtu E (2020) A better use of audio-visual cues: Dense video captioning with bi-modal transformer. In: Br Mach Vis Conf, pp 1–16
19. Estevam V, Laroca R, Pedrini H, Menotti D (2021) Dense video captioning using unsupervised semantic information, pp 1–12. arXiv preprint arXiv:2112.08455
20. Wang Q, Chen K (2017) Alternative semantic representations for zero-shot human action recognition. In: Machine Learning and Knowledge Discovery in Databases, pp 87–102. https://doi.org/10.1007/978-3-319-71249-9_6
21. 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: Int Conf Comput Vis, pp 2712–2719. https://doi.org/10.1109/ICCV.2013.337
22. Wu Z, Fu Y, Jiang Y, Sigal L (2016) Harnessing object and scene semantics for large-scale video understanding. In: Conf Comput Vis Pattern Recognit, pp 3112–3121. https://doi.org/10.1109/CVPR.2016.339

23. Gao J, Zhang T, Xu C (2019) I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. Conf Artif Intell 33:8303–8311. https://doi.org/10.1609/aaai.v33i01.33018303
24. Speer R, Chin J, Havasi C (2017) ConceptNet 5.5: An open multilingual graph of general knowledge. In: Conf Artif Intell, pp 4444–4451. https://doi.org/10.5555/3298023.3298212
25. Ghosh P, Saini N, Davis LS, Shrivastava A (2020) All about knowledge graphs for actions, pp 1–14. arXiv preprint arXiv:2008.12432
26. Kim TS, Jones JD, Peven M, Xiao Z, Bai J, Zhang Y, Qiu W, Yuille A, Hager GD (2021) DASZL: Dynamic action signatures for zero-shot learning. In: Conf Artif Intell, pp 1–10. https://doi.org/10.1609/aaai.v35i3.16276
27. Zhang B, Hu H, Sha F (2018) Cross-modal and hierarchical modeling of video and text. In: Eur Conf Comput Vis, pp 385–401. https://doi.org/10.1007/978-3-030-01261-8_23
28. Piergiovanni A, Ryoo MS (2020) Learning multimodal representations for unseen activities. In: Winter Conf Appl Comput Vis, pp 517–526. https://doi.org/10.1109/WACV45572.2020.9093612
29. Reimers N, Gurevych I (2019) Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Conf Empir Methods Nat Lang Process
30. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: International Conference on Neural Information Processing, pp 6000–6010
31. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Conference of the North American, Minneapolis, Minnesota, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423
32. 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
33. Krishna R, Hata K, Ren F, Fei-Fei L, Niebles JC (2017) Dense-captioning events in videos. In: Int Conf Comput Vis, pp 706–715. https://doi.org/10.1109/ICCV.2017.83
34. 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. https://doi.org/10.1109/ICCV.2011.6126543
35. Soomro K, Zamir AR, Shah M (2012) UCF101: A dataset of 101 human actions classes from videos in the wild, pp 1–6. arXiv preprint arXiv:1212.0402
36. Brattoli B, Tighe J, Zhdanov F, Perona P, Chalupka K (2020) Rethinking zero-shot video classification: End-to-end training for realistic applications. In: Conf Comput Vis Pattern Recognit, pp 4613–4623. https://doi.org/10.1109/CVPR42600.2020.00467
37. Roitberg A, Martinez M, Haurilet M, Stiefelhagen R (2018) Towards a fair evaluation of zero-shot action recognition using external data. In: Eur Conf Comput Vis Workshops, pp 1–9. https://doi.org/10.1007/978-3-030-11018-5_8
38. Gowda SN, Sevilla-Lara L, Kim K, Keller F, Rohrbach M (2021) A new split for evaluating true zero-shot action recognition. In: Ger Conf Pattern Recognit, pp 1–15
39. Mandal D, Narayan S, Dwivedi SK, Gupta V, Ahmed S, Khan FS, Shao L (2019) Out-of-distribution detection for generalized zero-shot action recognition. In: Conf Comput Vis Pattern Recognit, pp 9985–9993. https://doi.org/10.1109/CVPR.2019.01022
40. Lee J, Kim H, Byun H (2021) Sequence feature generation with temporal unrolling network for zero-shot action recognition. Neurocomputing 448:313–323. https://doi.org/10.1016/j.neucom.2021.03.070
41. Sun B, Kong D, Wang S, Li J, Yin B, Luo X (2022) GAN for vision, KG for relation: A two-stage network for zero-shot action recognition. Pattern Recognit. 126. https://doi.org/10.1016/j.patcog.2022.108563
42. Mohamed MA, Mertsching B (2012) TV-L1 optical flow estimation with image details recovering based on modified census transform. In: Int Symp Vis Comput, pp 482–491. https://doi.org/10.1007/978-3-642-33179-4_46
43. Sun D, Yang X, Liu M, Kautz J (2017) PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume, pp 1–18. arXiv preprint arXiv:1709.02371
44. Hershey S, Chaudhuri S, Ellis DPW, Gemmeke JF, Jansen A, Moore RC, Plakal M, Platt D, Saurous RA, Seybold B, Slaney M, Weiss RJ, Wilson K (2017) CNN architectures for large-scale audio classification. In: Int Conf Acoust Speech Signal Process, pp 131–135. https://doi.org/10.1109/ICASSP.2017.7952132
45. Gemmeke JF, Ellis DPW, Freedman D, Jansen A, Lawrence W, Moore RC, Plakal M, Ritter M (2017) Audio set: An ontology and human-labeled dataset for audio events. In: Int Conf Acoust Speech Signal Process, pp 776–780. https://doi.org/10.1109/ICASSP.2017.7952261
46. Banerjee S, Lavie A (2005) METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Workshop Intrinsic Extrinsic Eval Measures Mach Transl Summarization, pp 65–72

47. Papineni K, Roukos S, Ward T, Zhu W (2002) BLEU: a method for automatic evaluation of machine translation. In: Annu Meet Assoc Comput Linguist, pp 311–318. https://doi.org/10.3115/1073083.1073135
48. Reimers N, Gurevych I (2020) Making monolingual sentence embeddings multilingual using knowledge distillation. In: Conf Empir Methods Nat Lang Process, pp 4512–4525. https://doi.org/10.18653/v1/2020.emnlp-main.365
49. Xian Y, Akata Z, Sharma G, Nguyen Q, Hein M, Schiele B (2016) Latent embeddings for zero-shot classification. In: Conf Comput Vis Pattern Recognit, pp 69–77. https://doi.org/10.1109/CVPR.2016.15
50. Changpinyo S, Chao W-L, Gong B, Sha F (2016) Synthesized classifiers for zero-shot learning. In: Conf Comput Vis Pattern Recognit, pp 5327–5336. https://doi.org/10.1109/CVPR.2016.575
51. Xian Y, Lorenz T, Schiele B, Akata Z (2018) Feature generating networks for zero-shot learning. In: Conf Comput Vis Pattern Recognit, pp 5542–5551. https://doi.org/10.1109/CVPR.2018.00581
52. Gowda SN, Sevilla-Lara L, Keller F, Rohrbach M (2022) CLASTER: Clustering with reinforcement learning for zero-shot action recognition. In: Eur Conf Comput Vis, pp 187–203. https://doi.org/10.1007/978-3-031-20044-1_11
53. Gowda SN (2023) Synthetic sample selection for generalized zero-shot learning. arXiv preprint arXiv:2304.02846
54. Lampert CH, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: Conf Comput Vis Pattern Recognit, pp 951–958. https://doi.org/10.1109/CVPR.2009.5206594
55. Liu J, Kuipers B, Savarese S (2011) Recognizing human actions by attributes. In: Conf Comput Vis Pattern Recognit, pp 3337–3344. https://doi.org/10.1109/CVPR.2011.5995353
56. Xu X, Hospedales T, Gong S (2015) Semantic embedding space for zero-shot action recognition. In: Int Conf Image Process, pp 63–67. https://doi.org/10.1109/ICIP.2015.7350760
57. Romera-Paredes B, Torr PHS (2015) An embarrassingly simple approach to zero-shot learning. In: Int Conf Mach Learn, pp 2152–2161. https://doi.org/10.1007/978-3-319-50077-5_2
58. Akata Z, Reed S, Walter D, Lee H, Schiele B (2015) Evaluation of output embeddings for fine-grained image classification. In: Conf Comput Vis Pattern Recognit, pp 2927–2936. https://doi.org/10.1109/CVPR.2015.7298911
59. Xu X, Hospedales T, Gong S (2016) Multi-task zero-shot action recognition with prioritised data augmentation. Eur Conf Comput Vis 9906:343–359. https://doi.org/10.1007/978-3-319-46475-6_22
60. 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: Conf Comput Vis Pattern Recognit, pp 1042–1051. https://doi.org/10.1109/CVPR.2017.117
61. Zhu Y, Long Y, Guan Y, Newsam SD, Shao L (2018) Towards universal representation for unseen action recognition. In: Conf Comput Vis Pattern Recognit, pp 9436–9445. https://doi.org/10.1109/CVPR.2018.00983
62. Kerrigan A, Duarte K, Rawat Y, Shah M (2021) Reformulating zero-shot action recognition for multi-label actions. Conf Neural Inf Process Syst 34:25566–25577
63. Huang K, Miralles-Pechuán L, McKeever S (2022) Combining text and image knowledge with GANs for zero-shot action recognition in videos. In: Int Conf Comput Vis Theory Appl, pp 623–631. https://doi.org/10.5220/0010903100003124
64. Doshi K, Yilmaz Y (2022) End-to-end semantic video transformer for zero-shot action recognition. arXiv preprint arXiv:2203.05156
65. Bretti C, Mettes P (2021) Zero-shot action recognition from diverse object-scene compositions. In: Br Mach Vis Conf, pp 1–14
66. Pagliardini M, Gupta P, Jaggi M (2018) Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In: Conference of the North American, pp 528–540. https://doi.org/10.18653/v1/N18-1049

Authors and Affiliations

Valter Estevam[1,2] · Rayson Laroca[2] · Helio Pedrini[3] · David Menotti[2]

Rayson Laroca
rblsantos@inf.ufpr.br

Helio Pedrini
helio@ic.unicamp.br

David Menotti
menotti@inf.ufpr.br

1 Federal Institute of Paraná, Irati 84500-000, Paraná, Brazil

2 Department of Informatics, Federal University of Paraná, Curitiba 81531-970, Paraná, Brazil

3 Institute of Computing, University of Campinas, Campinas 13083-852, São Paulo, Brazil