Zero-Shot Action Recognition with ChatGPT-Based Instruction

Nan Wu(✉), Hiroshi Kera, and Kazuhiko Kawamoto

Chiba University, Chiba 263-8522, Japan
{gonan,kera}@chiba-u.jp, kawa@faculty.chiba-u.jp

Abstract. As deep learning continues to evolve, datasets are becoming increasingly larger, leading to higher costs for manual labeling. Zero-shot learning eliminates the need for substantial labor costs to label training datasets. It even allows the model to predict new classes with slight modifications. However, the accuracy of zero-shot learning still remains relatively low and makes practical applications challenging. Some recent research has tried to improve accuracy by manually annotating features, but this approach again requires labor-intensive input. To reduce these labor costs, we employ ChatGPT, which can generate features automatically without any manual involvement. Importantly, this approach maintains high accuracy levels, surpassing other automated methods.

Keywords: Zero-shot learning · Zero-shot action recognition · Large language models

1 Introduction

In the field of deep learning, to continuously improve the accuracy of deep models, people make the training dataset larger and larger. The advantage of this method is that the model can learn more knowledge, but it also increases the cost of labeling datasets and training models. This cost is difficult for ordinary people to afford. Especially in action recognition tasks, we usually classify videos instead of images. Videos have a larger amount of data than images, so training an action recognition model will cost more labor. In order to solve this problem, people began to study zero-shot learning. With zero-shot action recognition (ZSAR), the model can recognize a new action even if the model has never seen the action.

However, the accuracy of zero-shot action recognition is still low, and it is difficult to apply it in practice. Wu et al. [1] propose the method of manually annotating features to improve accuracy, but this method requires labor costs. With this method, people have to manually annotate several text features for each action. The more features, the higher the accuracy. In order to reduce labor cost, we choose to use ChatGPT to automatically generate features without any manual operation. ChatGPT is a model improved by InstructGPT [2]. It can generate specified text content according to the request of users. In this paper,

B. Xin et al. (Eds.): IWACIII 2023, CCIS 1932, pp. 18–28, 2024.
https://doi.org/10.1007/978-981-99-7593-8_3

we will let the model generate several text features for each action, and then use these features to instruct action recognition. This method can remove the labor cost in [1]. At the same time, it can keep the accuracy at a high level, surpassing other automatic methods.

2 Related Work

2.1 Zero-Shot Learning

With zero-shot learning, a model can recognize an object even if the model has never seen its class. Although the model does not need the training dataset of this class, it still needs some features to represent this class.

The work [3] explores zero-shot learning with deep learning models early. It used images of animals as input and animal features such as skin color and body size as output. Although this model can not classify the type of animal, it can identify its features. After obtaining these features, the class of the animal images can be predicted. This process does not require images of new animal classes for training, just their features. During testing, the model will output all the features of an input animal image. These are then compared with known animal features to find the closest match and label the image accordingly. However, this method requires manual production of features, leading to labor costs.

Frome et al. [4] improved the method to use the word embedding of the animal name as the output instead of the features of the animal. This method makes zero-shot learning more automated and reduces labor costs. Xu et al. [5] used compact watershed segmentation to divide the image into multiple small images, which were put into the CNN to cluster the output into multiple classes. They treated each class as a feature. These features were used to infer the class of the image. Cheng et al. [6] proposed a hybrid routing transformer (HRT) model. In the HRT encoder, we embed active attention, which is constructed by both the bottom-up and the top-down dynamic routing pathways to generate the attribute-aligned visual feature. The author of [7] also used contrastive embedding to exploit both the class-wise and instance-wise supervision for generalized zero-shot learning, under the attribute-guided weakly supervised representation learning framework.

2.2 Zero-Shot Action Recognition

In action recognition tasks, videos are usually used instead of images. Videos have more data than images, leading to higher annotation and training costs. In order to solve this problem, zero-shot learning was introduced for action recognition. With zero-shot action recognition (ZSAR), the model can recognize new actions that have never been seen before. The method of ZSAR is shown in Fig. 1. Kerrigan et al. [8] transformed videos and labels into vectors and multiplied the vectors of the videos and labels. Then, they put the product into a fully connected network, and the final output represented the matching degrees

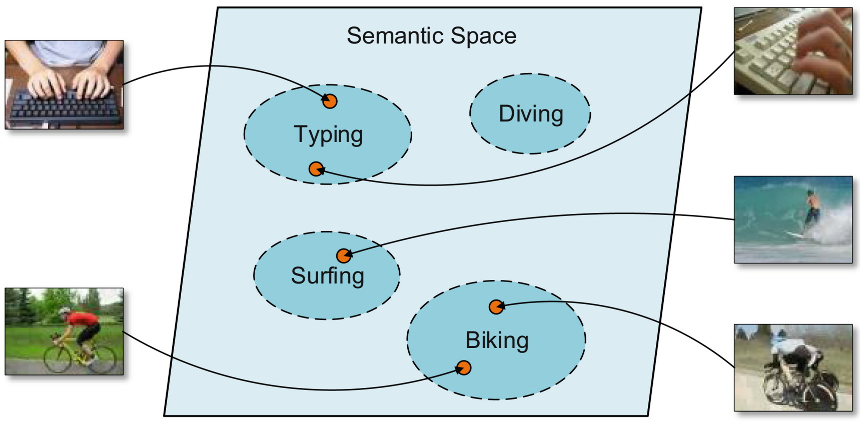

Fig. 1. Zero-shot action recognition.

between the video and labels. Similarly, VD-ZSAR [9] transformed videos and labels into vectors, which in turn were transformed into a new visual-semantic joint embedding space to identify the most appropriate label for the video in the new space.

2.3 Improving Zero-Shot Learning with Text

In order to improve the accuracy of zero-shot learning, commonly used methods include optimizing the feature extraction module and improving the accuracy of mapping from visual space to semantic space. A common semantic space is usually the word embedding of the label. But the label, usually one word, contains too little information, usually only a few words, so it is prone to errors. So people began to use long sentences or a paragraph to represent the action, which can improve the stability of semantic space mapping. [10] obtains descriptions of actions from Wikipedia and manually filters out suitable content. Then the authors detect the object of the frames and convert the names of the objects into feature vectors. The position of the action in the semantic space is determined by these two parts. [11] uses a template—"This is a video about Attributes" to generate a sentence. The sentence is then converted into a feature vector with CLIP [12]. The feature vector and the word embedding of the label are used together to predict the class of the video.

2.4 Large Language Models

In recent advancements in computer hardware, language models have grown in size, leading to the development of Large Language Models (LLMs) such as GPT-3 [13] and PaLM [14]. These models are trained on large datasets that mainly consist of text from the web. As a result, LLMs store a wide range of

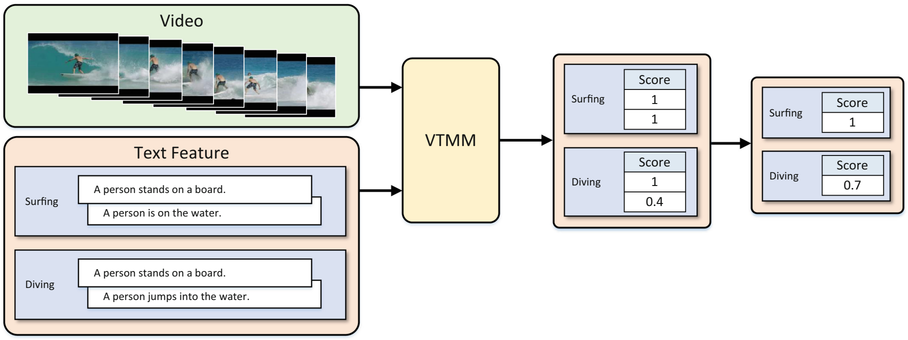
Video
Text Feature
Surfing
A person stands on a board.
A person is on the water.
Diving
A person stands on a board.
A person jumps into the water.
VTMM
Surfing
Score
1
1
Diving
Score
1
0.4
Surfing
Score
1
Diving
Score
0.7

Fig. 2. Use VTMM for classification. VTMM has two inputs, one is the video and the other is the text features. The model can determine whether the video and text features match. We can use VTMM to obtain the features of a video, and then predict the class of the video based on this information.

knowledge, making people useful tools for retrieving information. Naeem et al. [15] have employed the knowledge stored in LLMs to improve the accuracy of image classification tasks. In the visual question answering task, Guo et al. [16] extracts various features from the image and converts them into text contents. The text contents are then put into LLM and finally the answer is obtained.

3 Method

We choose [1] as our baseline. The authors of [1] propose a video-text matching model (VTMM), which can determine whether the video and text features match. The authors used manually annotated text features to predict the classes of videos. In this section, we use ChatGPT to automatically generate text features to replace the features in the baseline.

3.1 Video-Text Matching Model

The method of action recognition using VTMM is shown in Fig. 2. This model has two inputs, one is the video and the other is the text features. These text features are used to describe what the person in the video is doing. The model can determine whether the video and text features match. The output of the model takes a value in the range $[0, 1]$. If the video matches the features, then the output will be close to 1. Otherwise, the output will be close to 0. For example, if the input is a video of *BrushingTeeth*, and the text feature is "A person is holding a toothbrush in the hands". If the text feature is replaced with "A person is swimming in water", then the matching degree will be close to 0. For the action of *HighJump*, we can make three features—"A person runs", "A person jumps up", and "A person falls on the mat". If a video has these three features, then it probably belongs to the action of *HighJump*. Conversely,

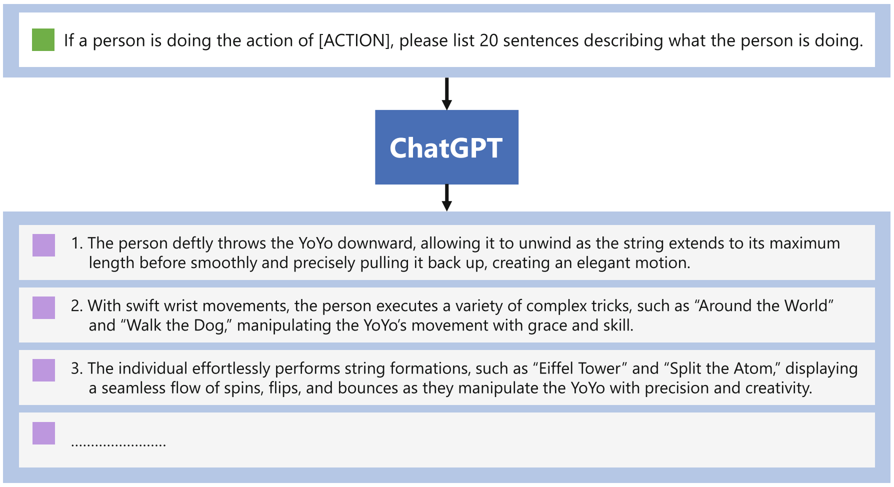
If a person is doing the action of [ACTION], please list 20 sentences describing what the person is doing.
ChatGPT
1. The person deftly throws the YoYo downward, allowing it to unwind as the string extends to its maximum length before smoothly and precisely pulling it back up, creating an elegant motion.
2. With swift wrist movements, the person executes a variety of complex tricks, such as "Around the World" and "Walk the Dog," manipulating the YoYo's movement with grace and skill.
3. The individual effortlessly performs string formations, such as "Eiffel Tower" and "Split the Atom," displaying a seamless flow of spins, flips, and bounces as they manipulate the YoYo with precision and creativity.
....

Fig. 3. Use ChatGPT to automatically generate text features for actions.

if there is no feature or only one feature in the video, then this video is likely to be another action. With this method, we can let the model recognize a new action without using video data to train the model. We calculate the matching degree between the video and action class:

$$s = \frac{1}{k}\sum_{i=1}^{k} s_i, \tag{1}$$

where s_i is the matching degree between the text feature and the video, and s is the matching degree between the action class and the video.

3.2 Automatically Generate Text Features

In [1], the author manually annotated text features for each action. People watch the video, analyze the features of the video, and then express it in text. This work takes manpower, which is also a cost. In order to reduce labor costs and make the model automatic, we use ChatGPT to generate features. The method is shown in Fig. 3.

We take the action YoYo as an example to list different prompts and the content of ChatGPT replies. If we ask "Please list three features of YoYo", then ChatGPT replies as follows.

1. Hand-eye coordination: YoYo requires precise hand-eye coordination as the performer needs to control the YoYo's movements while keeping a visual track of its position relative to the string.
2. Fine motor skills: The manipulation of the YoYo involves intricate finger movements, requiring dexterity and fine motor skills. These skills are developed and honed through practice and can have benefits beyond YoYo play.

3. Focus and concentration: Successfully executing YoYo tricks demands concentration and focus. The performer needs to maintain mental clarity and be fully present to anticipate and react to the YoYo's movements in real-time. This can enhance focus and improve mental agility.

These responses are not related to the appearance features of YOYO; therefore, these features can not be recognized by the VTMM model.

Next, to let ChatGPT answer the appearance features, we modify the prompt to "Please list three sentences describing the features of the action YoYo from the appearance.". The modification yields the following responses.

1. The action YoYo boasts a sleek and aerodynamic design with a precision-engineered aluminum body, allowing for smooth and effortless spins and tricks.
2. Its responsive ball-bearing system ensures quick returns and optimal control, enabling users to perform intricate string maneuvers and achieve impressive speeds.
3. The YoYo is equipped with a customizable string length and a wide gap between the halves, facilitating advanced string tricks and making it suitable for both beginners and experienced YoYo enthusiasts.

We can see that the output features are mainly some appearance features, but several subjective evaluations still remain.

After trial and error, we found a better one—"If a person is doing the action of YoYo, please list 3 sentences describing what the person is doing." The responses are as follows.

1. The person deftly throws the YoYo downward, allowing it to unwind as the string extends to its maximum length before smoothly and precisely pulling it back up, creating an elegant motion.
2. With swift wrist movements, the person executes a variety of complex tricks, such as "Around the World" and "Walk the Dog," manipulating the YoYo's movement with grace and skill.
3. The individual effortlessly performs string formations, such as "Eiffel Tower" and "Split the Atom," displaying a seamless flow of spins, flips, and bounces as they manipulate the YoYo with precision and creativity.

We change this sentence into a template—"If a person is doing the action of ____, please list 20 sentences describing what the person is doing." We use this prompt template to generate 20 features for each action. We set the weight of each feature to 1.

3.3 Different Prefixes of Text Feature

Referring to [23], if different prefixes are added in front of the text, then the accuracy will also change. After adding the prefix, the meaning of the sentence remains the same. So we try to modify the text features. "A photo of", "A video of" and "As the photo shows," are added before the text features. We use a total of 4 different types of text features to evaluate the accuracy of the model. Finally, the one with the highest accuracy is selected as the best text feature.

4 Experiment

In this section, we will evaluate our model and compare it with other models. The effect of hyperparameters on performance is then also tested.

4.1 Compare with Other Models

We choose UCF101 [17] and HMDB51 [18] as our dataset. UCF101 had 13,320 videos and 101 actions. The 101 classes are randomly divided into 51 seen and 50 unseen classes. HMDB51 had 7,000 videos and 51 actions. The 51 classes are randomly divided into 26 seen and 25 unseen classes.

We use ChatGPT to generate 20 text features for each action, and "A photo of", "A video of" and "As the photo shows," are added before the text features. We use a total of 4 different types of text features and the one with the highest accuracy is selected as the best text feature. We use these features to classify the video and evaluate the accuracy of the model. The results are shown in Table 1. From the results, we can see that our method improves the automation of the model at the expense of a small amount of accuracy. Although the correct of our model is lower than baseline, it is still higher than other models. Among the models that do not require human cost, our model achieves the best results.

4.2 The Effect of Different Prompts

We tried dozens of prompts, and due to time constraints, we could not use all prompts to calculate the accuracy. Three representative prompts were selected for experiments. We use these three different prompts to generate 20 text features for each action of UCF101, then use these text features to classify the video and calculate the accuracy. Finally, the results are obtained, as shown in Table 2. It can be seen that the accuracy of prompt 3 is the highest. However, in practice, the prompts have little effect.

4.3 The Effect of Different Text Feature

In this section, we evaluate the impact of different prefixes of text features on accuracy. These sentences have the same meaning but use different expressions. We try to modify the text features of UCF101 and HMDB51. "A photo of", "A video of" and "As the photo shows," are added before the text features, and the results are shown in Table 3. We can see that modifying the format of the prompt will affect the model. Modifications will increase or decrease the accuracy.

4.4 The Effect of the Number of Features

In this section, we test the effect of the number of features on accuracy. We first use ChatGPT to generate 20 features for each action of UCF101. Then we randomly select 1 feature from 20 features and use it to predict the classes of

Table 1. Accuracy of each model and the best results are in bold. We used ChatGPT to generate 20 text features for each action and used these features to classify the video and evaluate the accuracy of the model.

Model	Reference	UCF101	HMDB51
DAP	CVPR2009 [3]	14.3	N/A
IAP	CVPR2009 [3]	12.8	N/A
Two-Stream GCN	AAAI2019 [19]	33.4	21.9
BD-GAN	Neurocomputing2020 [20]	18.7	23.9
VDARN	Ad Hoc Networks2021 [21]	26.4	21.6
ER	ICCV2021 [10]	51.8	35.3
LUPI	ACCV2022 [22]	52.6	38.8
VTMM + ChatGPT	Ours	**72.9**	**59.3**
VTMM + handcrafted features	arXiv [1]	**78.1**	**59.4**

Table 2. The accuracy of different prompts. Three representative prompts was selected for experiments. We used these three different prompts to generate 20 text features for each action of UCF101, then used these text features to classify the video and calculate the accuracy.

Prompt	Prompt Content	Accuracy
1	Please list 20 features of action ____	69.3
2	Please list 20 sentences describing the features of the action ____ from the appearance	68.1
3	If a person is doing the action of ____ , please list 20 sentences describing what the person is doing	69.5

videos. In order to reduce the error caused by randomness, the same experiment was repeated 30 times, and finally, the average value of the accuracies was calculated. Then, we gradually increase the number of features used, from 1 to 20. The same experiment was repeated 30 times. Finally, the relation between the number of features and the accuracy was obtained, as shown in Fig. 4. From the result, we can see that the more features, the higher the accuracy. However, after the number of features exceeds 10, the growth rate of the accuracy slows down.

Table 3. The accuracy of different features. The best results are in bold. We add different text in front of the features of UCF101 and HMDB51, we can see that modifying the format of the prompt will affect the accuracy of the model.

Feature Content	UCF101	HMDB51
[FEATURE]	69.5	56.4
A photo of [FEATURE]	**72.9**	58.7
A video of [FEATURE]	67.1	57.0
As the photo shows, [FEATURE]	71.5	**59.3**

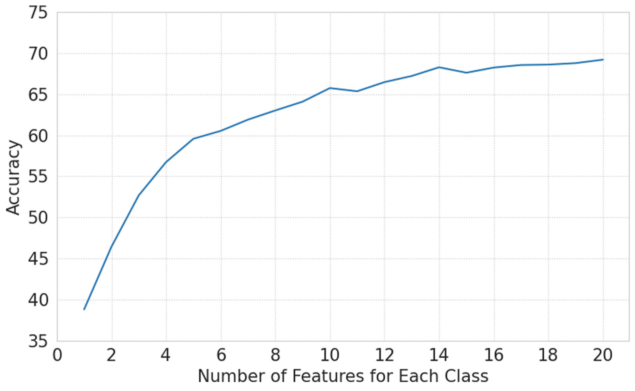

Fig. 4. The effect of the number of features. We use ChatGPT to annotate each action with a varying number of text features, and then use these features to classify videos. We can see that the more text features, the higher the accuracy of action recognition. However, the growth rate of the accuracy slows down.

5 Conclusion

The accuracy of zero-shot action recognition is still low, and it is difficult to apply it in practice. There are also some recent studies that use the method of manually labeling features to improve accuracy, but this method requires labor costs. In order to reduce labor costs, we choose to use ChatGPT to automatically generate features without any manual operation. We use ChatGPT to generate 20 features for each action and use these features to instruct the model to recognize actions. After experiments, our model has achieved the highest accuracy among models that do not require labor costs. Compared with the baseline, our model removes the labor cost at the expense of a small amount of accuracy. This method is affected by the prompt. If we want to continue to improve the accuracy, then we need to find a better prompt or use feedback to optimize the prompt.

Funding. This work was supported by the Japan Society for the Promotion of Science KAKENHI Grant Number JP19K12039 and JP22H03658.

References

1. Wu, N., Kera, H., Kawamoto, K.: Improving zero-shot action recognition using human instruction with text description. Appl. Intell. 1–15 (2023). https://doi.org/10.1007/s10489-023-04808-w
2. Ouyang, L., et al.: Training language models to follow instructions with human feedback. Proc. NeurIPS **35**, 27730–27744 (2022)
3. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Proceedings of CVPR, pp. 951–958 (2009)
4. Frome, A., et al.: Devise: A deep visual-semantic embedding model. In: Proceedings of NeurIPS, pp. 2121–2129 (2013)
5. Xu, W., Xian, Y., Wang, J., Schiele, B., Akata, Z.: VGSE: Visually-grounded semantic embeddings for zero-shot learning. In: Proceedings of CVPR, pp. 9316–9325 (2022).
6. Cheng, D., Wang, G., Wang, B., Zhang, Q., Han, J., Zhang, D.: Hybrid routing transformer for zero-shot learning. Pattern Recogn. **137**, 109270 (2023)
7. Cheng, D., Wang, G., Wang, N., Zhang, D., Zhang, Q., Gao, X.: Discriminative and robust attribute alignment for zero-shot learning. IEEE Transactions on Circuits and Systems for Video Technology (Early Access) **33**(8), 4244–4256 (2023)
8. Kerrigan, A., Duarte, K., Rawat, Y., Shah, M.: Reformulating zero-shot action recognition for multi-label actions. Proc. NeurIPS **34**, 25566–25577 (2021)
9. Xing, M., Feng, Z., Su, Y., Peng, W., Zhang, J.: Ventral and dorsal stream theory based zero-shot action recognition. Pattern Recogn. **116**, 107953 (2021)
10. Chen, S., Huang, D.: Elaborative rehearsal for zero-shot action recognition. In: Proceedings of ICCV, pp. 13638–13647 (2021)
11. Wu, W., Wang, X., Luo, H., Wang, J., Yang, Y., Ouyang, W.: Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6620–6630 (2023)
12. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
13. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
14. Chowdhery, A., et al.: Palm: Scaling language modeling with pathways (2022). arXiv preprint arXiv:2204.02311
15. Naeem, M.F., et al.: I2MVFormer: large language model generated multi-view document supervision for zero-shot image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15169–15179 (2023)
16. Guo, J., et al.: From images to textual prompts: zero-shot visual question answering with frozen large language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10867–10877 (2023)
17. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild (2012). arXiv:1212.0402
18. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: Proceedings of ICCV, pp. 2556–2563 (2011)

19. Gao, J., Zhang, T., Xu, C.: I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. Proc. AAAI **33**(1), 8303–8311 (2019)
20. Mishra, A., Pandey, A., Murthy, H.A.: Zero-shot learning for action recognition using synthesized features. Neurocomputing **390**, 117–130 (2020)
21. Su, Y., Xing, M., An, S., Peng, W., Feng, Z.: Vdarn: Video disentangling attentive relation network for few-shot and zero-shot action recognition. Ad Hoc Netw. **113**, 102380 (2021)
22. Gao, Z., Hou, Y., Li, W., Guo, Z., Yu, B.: Learning using privileged information for zero-shot action recognition. In: Proc. ACCV, pp. 773–788 (2022)
23. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. Int. J. Comput. Vision **130**(9), 2337–2348 (2022)