From Segment Localization to Natural Language Answers

Lorenzo Calogiuri
Politecnico di Torino
s334186@studenti.polito.it

Minal Jamshed
Politecnico di Torino
s329091@studenti.polito.it

Prima Acharjee
Politecnico di Torino
s329198@studenti.polito.it

Abstract

Egocentric videos, which capture everyday activities from a first-person perspective, offer unique challenges and opportunities for leveraging video contents through Natural Language Queries. Traditional models for video segment localization output time intervals in response to queries, requiring viewers to watch these segments to obtain the actual answers. Differently, we propose a pipeline in which relevant video segments are identified using advanced video segment localization models, and then these segments are processed by a VideoQA (Video Question Answering) Model to obtain natural language answers. This approach efficiently handles long videos by focusing computational resources on relevant segments, thus processing queries efficiently. The proposed pipeline effectively addresses the challenge, exhibiting an average improvement of +3.2% over the segment localization baseline and demonstrating optimal performance in generating natural language responses.

1. Introduction

Unlike traditional third-person videos, egocentric videos offer insights into the interactions of the wearer, activities, and environmental context from their point of view. Egocentric vision refers to visual data captured from a first-person perspective, usually using wearable cameras. This presents a continuous and immersive view of the world as experienced by the camera wearer [11]. The challenges associated with egocentric vision include dealing with lengthy video data, occlusions caused by the wearer's body, and the requirement to understand complex and dynamic scenes in real-time.

This project has used narrated videos from the Ego4D dataset, which is a massive egocentric Machine Learning video dataset comprising 3670 hours of narrated videos from first-person point of view. The videos cover various daily life activities capturing the natural habitats of 931 individuals from 74 different locations of the world [5]. The Ego4D dataset is a great resource for training and assessing models in tasks like action recognition, object detection, and video summarization. The goal of this large-scale egocentric dataset is to further enhance first-person vision research. It seeks to solve the particular difficulties presented by egocentric videos and to encourage the creation of new models and algorithms specifically for this field.

The focus of this project is based on the Ego4D benchmark for past episodic memories, which aims to understand and retrieve specific events from past experiences captured in egocentric videos [1]. This benchmark comprises of three types of challenges namely: Moments Queries (MQ), which locate and return each instance of a specified activity within the video; Visual Queries (VQ), which use an image to find an object in the video and provide both temporal and spatial details; and Natural Language Queries (NLQ), which involve text-based requests for specific past events and yield corresponding video segments. Our project primarily focuses on the Natural Language Queries challenge within the Episodic Memory Benchmark of Ego4D [3]. NLQs are used to localize video segments that answer a given natural language question (e.g. "What did I put in the plate?"). Traditional models for this task output time intervals, requiring viewers to watch the segments to obtain answers. This approach is often very time-consuming and involves processing lengthy video segments [1].

Our proposed pipeline leverages these challenges by efficiently generating natural language responses to egocentric video queries. During implementation, we undertake several steps to improve the processing of Natural Language Queries within the Ego4D dataset. We trained several models using pre-extracted features to identify the best performing model compared to established baselines. Leveraging this optimized model, we extracted video segments that provided the most accurate predictions for natural language query responses. These selected video intervals were then processed through a video question-answer model to generate natural language answers corresponding to the NLQ. By transitioning

from identifying relevant video intervals to extracting textual answers, we aim to streamline the video question-answering pipeline, reducing the segments that the VideoQA model need to process and thereby increasing overall efficiency [9].

2. Related Works

Development of Egocentric Task Verification (EgoTV) Benchmark. It aims to verify the execution of tasks from egocentric videos based on natural language descriptions. This includes developing a synthetic dataset and a novel Neuro-Symbolic Grounding (NSG) approach for task verification. The prior work on EgoTV is about verifying the correctness of performed tasks in videos against a task description, requiring detailed understanding and reasoning about actions and their sequences [6]. In contrast, we centered on localizing and retrieving segments in a video that correspond to a specific natural language query, focusing more on relevant segments and retrieval of video content based on language query rather than verifying task performance.

Assessing goal-oriented tasks in egocentric videos through multi-agent beliefs. The EgoTaskQA paper aims to establish a benchmark for assessing task comprehension via question-answering using annotated egocentric videos. It emphasizes grasping actions, intentions, goals, and interactions among multiple agents [8]. In contrast, our project aims to use natural language queries to identify relevant video segments from the presented lengthy videos.

Merge visual representation into the language feature space through an LVLM. This approach, demonstrated by the VideoQA model, allows it to learn multi-modal interactions from a combined dataset of images and videos, thereby improving performance across various image and video benchmarks. This model is trained on both images and videos and learns to interpret these modalities through a unified projection layer, enhancing performance across diverse multimedia and language comprehension benchmarks [9]. Our project involves training models like VSLBase and VSLNet on the Ego4D Episodic Memory benchmark pertaining to Natural Language Queries, using pre-extracted Omnivore and EgoVLP features. By focusing on transitioning from identifying video intervals to extracting textual answers, our goal is to reduce the segments to be processed by the videoQA model.

Identifying the relevant temporal window in egocentric videos. The work is rooted mostly in enhancing feature extraction and grounding effectiveness. It has implemented models like VSLNet and EgoVLP to improve the grounding of queries in video content. Specifically, it discusses the use of EgoVLP, a pre-trained feature extractor, and VSLNet, which is designed to enhance the interaction between video and text features for better localization of relevant video segments [7]. Essentially, our project builds on identifying useful segments to answer the queries efficiently, involving VLM post-segment identification to produce textual answers.

Architectural and Functional Analysis of VSL Models. Both VSLBase [16] and VSLNet [15, 16] models use fixed feature extractors for handling textual and visual inputs during training, with VSLBase using a 3-D CNN for video and GloVe embedding [16] for text, ensuring robust initial feature extraction. Moreover, the addition of the Query-Guided Highlighting (QGH) module in VSLNet represents a significant improvement. This module enhances the accuracy of video representation by emphasizing segments relevant to the query, thereby improving the model's ability to precisely localize video segments that answer specific queries. This added functionality leads to the superiority of VSLNet over VSLBase by providing a more refined and context-sensitive representation. The advancements in VSLNet over VSLBase through the QGH module demonstrate a strategic improvement in handling video segment localization tasks [16]. This positions VSLNet as potentially more adept for applications that demand high precision in video understanding and query response. Figure 1 shows the architectures of VSLBase and VSLNet, highlighting the difference between the two models.

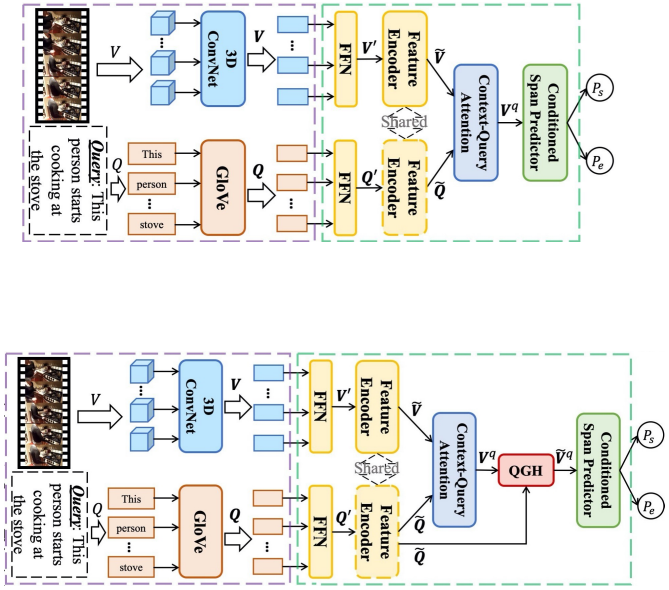

Figure 1. Comparison between the VSLBase (upper) and VSLNet (lower) architectures

3. Methodology

3.1. Training VSLBase and VSLNet

A. *From VSLNet to VSLBase*

Our project pipeline is initiated by cloning the official Ego4D Episodic Memory Repository, which includes specifically for the NLQ task, the VSLNet architecture [5]. As highlighted above, the major difference between VSLNet and VSLBase is the QGH module, therefore, by removing the Highlight Layer from the architecture of VSLNet, we can transform the model into VSLBase. With this modification, the model does not extend the segment localization boundary, considering only the target moment. The highlighting score, which is a scaling factor used to update features, is no longer calculated, as is the loss function that takes into account the presence of the QGH module [16].

B. *Training the models*

Downloading Annotations and Features. We attained the AWS license in order to access the Ego4D Dataset. Then we had set-up the CLI environment and downloaded the dataset in the Colab environment. After that, thoroughly explored the dataset by visualizing some of the videos and annotations for better understanding of the dataset (using the Official Ego4D Visualizer). The Ego4D metadata json files size is 2.51 GB. We also computed some statistical measures over the query duration to develop an overall understanding of the dataset, such as the relative query size, which is a value between 0 and 1 and computed as the ratio between the duration of the answer within the query and the query duration itself. Figure 2 shows a frequency histogram based on this measure.

Training models from scratch on video datasets often requires long computational times. To circumvent this problem, we leveraged the pre-trained Omnivore [4] and EgoVLP [17] models and fine-tuned them for the specific episodic memory task (NLQ) we aimed to solve. It should be noted that we used the same annotation data for training the models with Omnivore and EgoVLP features.

Omnivore features can be directly downloaded through the CLI using the `--datasets omnivore_video_swinl_fp16` option. This command downloads 1260 feature files having an overall size of 10.3 GB. We downloaded the EgoVLP features and placed them in the appropriate folder. The provided compressed folder contains 9611 feature files and has an overall size of 12.58 GB.

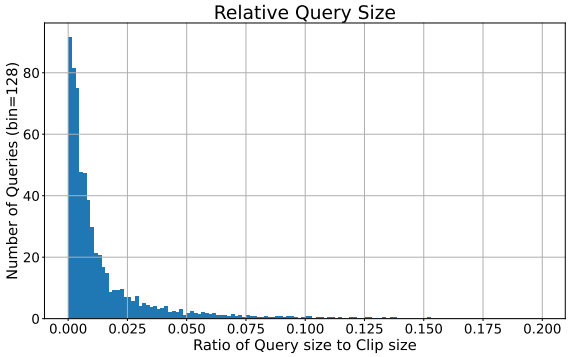

Figure 2. Frequency histogram of relative query size for queries having relative size less than 0.2

Cloning the Episodic Memory Baseline Repository. We cloned the official Ego4D Episodic Memory Repository [5] from GitHub to gain access to VSLNet and train the architecture on the extracted features and annotations. For VSLBase, we forked the Official Repository and made the modifications mentioned earlier.

Preparing the Dataset. The dataset preparation includes initializing the environment variables, which consist of the model name, features, and feature directories necessary for processing NLQ queries. This setup is crucial to ensure that all components required for running the NLQ task with VSLBase and VSLNet using Omnivore or EgoVLP features are properly configured. These variables facilitate efficient access to necessary features and ensure that all software dependencies are met.

Training and Evaluation. The Ego4d annotations are prepared for the training phase, allowing the model (VSLBase/VSLNet) to be trained. The annotation entries are subsequently divided into training (11,291), validation (3,874), and testing (4,004) sets respectively.

We used the default official Colab notebook hyperparameters when training both models (VSLBase and VSLNet) with both pre-extracted features (Omnivore and EgoVLP) in order to achieve satisfactory results while keeping computational times compatible with the free Colab version runtime constraints. However, the major change we made while using Omnivore and EgoVLP features includes tweaking the dimensions of their video features. We set the video feature dimension for Omnivore at 1536 and for EgoVLP at 256.

3.2. Replacing the Text Encoder of VSLNet

In this step, we are asked to implement a variation of the VSLNet architecture where we replace the default *Bidirectional Encoder Representations from Transformers* (BERT)

Encoder [16] with *Global Vectors for Word Representation* (GloVe) [14, 16] embeddings in the VSLNet architecture to analyze how the VSLNet model performs with both Omnivore and EgoVLP features.
We downloaded the file `glove.840B.300d.txt` from the official GloVe NLP GitHub Repository and placed it in the proper directory. Our model required the specific GloVe file with the following features:

- **840B:** 840 billion tokens (words) from the dataset were used to train the model.
- **300d:** Each word is represented as a vector with 300 dimensions.

The Episodic Memory repository, which includes the VSLNet architecture, uses BERT as the default text encoder for model training. However, the repository's code supports both BERT and GloVe encoders, depending on the specific application. Therefore, we modified the model's training script in Colab to switch the default text predictor encoder from BERT to GloVe to suit our needs.

We gathered the results to more effectively evaluate which model variation performs optimally, with the intention of implementing the most effective model in the final stage of our pipeline. These results are detailed in the Experiments section.

3.3. From Video Intervals to Textual Answers

From the predictions obtained in 4.1 and for each query of the validation set, we found the best prediction among the ones in list of predictions (the one with the highest IoU [12] with respect to the ground truth). We dropped the queries if the duration of the predicted interval was more than 10 seconds. This was done due to the limited computational resources present in Colab when dealing with the further steps of the extension (cutting and generating natural language answers). We created a handy data structure (list of Dictionaries) containing the queries along with the corresponding video_uid, clip_uid, the prediction interval, the textual question, the ground truth interval and other useful information.

We then proceeded by selecting the top 50 queries for which the video segment is correctly retrieved. We have manually annotated the natural language ground truth associated with the selected queries by watching the prediction interval for each query, creating our small validation set. We have installed the `boto3` package [13] to download the videos related to the previously retrieved video_uid corresponding to the best 50 selected queries using the `manifest.csv` file present in the annotations folder of the Ego4D Dataset. The `manifest.csv` contains the `S3` bucket link for downloading the videos directly from the AWS framework. We cut the videos and extracted the parts of interest, namely the prediction interval for each query using `ffmpeg`. In order to avoid problems when processing the cut videos, we re-encoded them in `H.264` for the video and AAC for the audio.

Lastly, we have adopted a *Video Question Answering model* (VideoQA) to feed the previously extracted parts along with the input queries, requesting for a natural language answer. In particular, we have decided to adopt the Video-LLaVA model [9]. We cloned the official Video-LLaVA repository and loaded the Video-LLaVA-7B pretrained model. This model is made of `7 billion` parameters and it has had international resonance for its capacity of handling complex tasks on large datasets and different scenarios. We evaluated the quality of the answers obtained through Video LLaVA using the following metrics. **`METEOR`** (Metric for Evaluation of Translation with Explicit Ordering) [2] is a method that evaluates the similarity between the predicted text and the ground truth by counting matching words, including synonyms, and considering the word order. Scores range from 0 to 1, with 1 indicating a perfect translation and 0 indicating no overlap in meaning or order. Moreover, **`ROUGE`** Precision Score (Recall-Oriented Understudy for Gisting Evaluation) [10] measures the longest common sub-sequence (LCS) between the predicted text and the ground truth. LCS refers to the set of words that appear in the same order in both texts. A higher precision score indicates a higher level of accuracy in the predicted text.

4. Experiments

4.1. Model Training of VSLBase and VSLNet

As mentioned previously, we trained both VSLBase and VSLNet using pre-extracted features of Omnivore and EgoVLP, using the default official Colab notebook hyperparameters with number of Epochs = 10 and the Learning Rate = 0.0025. Table 1 highlights the results obtained.

Table 1. Performance Metrics for VSL Models

Models	IoU = 0.3%		IoU = 0.5%	
	r@1	r@5	r@1	r@5
VSLBase + Omnivore	6.14	12.93	3.64	8.26
VSLBase + EgoVLP	7.33	14.61	4.52	9.68
VSLNet + Omnivore	6.48	14.04	3.72	8.70
VSLNet + EgoVLP	8.03	15.82	4.75	10.12

VSLNet with EgoVLP features performs the best across all the metrics compared to other model and features combinations. This model shows an improved score of "Rank@1" and "Rank@5" depicting a substantial gain for both IoU@0.3 and IoU@0.5, suggesting better localization accuracy [12].

Table 2. Comparison to Baseline

Models	IoU = 0.3%		IoU = 0.5%	
	r@1	r@5	r@1	r@5
VSLNet (Baseline)	5.45	10.74	3.12	6.63
VSLNet + EgoVLP	8.03	15.82	4.75	10.12

VSLNet with EgoVLP model significantly outperforms the baseline VSLNet model on the validation set. For instance, "Rank@1" improved from 5.45 in the baseline to 8.03, and "Rank@5" from about 10.74 to 15.82, in our implemented version. These results exhibit an average improvement of **+3.2%** over the segment localization baseline. Table 2 shows the results obtained in comparison to the baseline.

To summarize, VSLNet with EgoVLP features indicate a superior performance compared to the baseline and other feature/model combinations. This suggests that:

- EgoVLP features provide richer or more relevant information for video language tasks in this dataset compared to Omnivore. One of the explainations is that EgoVLP was previously trained on a sub portion of Ego4D.
- VSLNet architecture, particularly with EgoVLP features, effectively captures the necessary temporal and semantic relationships better than the baseline models and configurations. This effectively shows that the QGH module, which extends the prediction boundary, can cover additional contexts and also help the network to focus on subtle differences between video frames.

4.2. Model Training of a Variation of VSLNet with GloVe

Given that the VSLNet model performed better than VSLBase model, we used it as a foundational model to implement the changes from BERT to GloVe. Table 3 shows the performance of GloVe embeddings with both Omnivore and EgoVLP features.

It can be seen that there is a slight improvement in results when using GloVe embedding with EgoVLP features compared to Omnivore features. Therefore, to further explore this combination, we experimented by changing the number of epochs and learning rates of the model. Table 4 shows how varying number of epochs and learning rates impacted the results.

Table 3. GloVe vs BERT

VSLNet Model	IoU = 0.3%		IoU = 0.5%	
Epoch = 10, LR = 0.0025	r@1	r@5	r@1	r@5
EgoVLP + BERT	8.03	15.82	4.75	10.12
EgoVLP + GloVe	3.79	9.53	2.17	5.83
Omnivore + GloVe	3.05	8.36	1.21	4.34

Table 4. Model Performance Across Different Settings

VSLNet	IoU = 0.3%		IoU = 0.5%	
EgoVLP + GloVe	r@1	r@5	r@1	r@5
E = 10, LR = 0.005	0.44	4.39	0.15	2.61
E = 10, LR = 0.0025	3.79	9.53	2.17	5.83
E = 10, LR = 0.0005	1.78	5.94	1.01	3.20
E = 30, LR = 0.005	0.44	5.14	0.15	3.15
E = 30, LR = 0.0025	7.54	15.23	4.70	10.22

It can be observed that by changing the number of epochs and the learning rate of the model provided no significant improvement in the performance of the model using GloVe embedding.

Therefore, from the results presented above, it is evident that the model using the BERT encoder performed better than the one with GloVe embeddings. We identified several reasons for this result:

- BERT is adept at grasping the context of word usage because it analyzes the full sequence of words within a sentence or across multiple sentences to understand each word's meaning, showing a better contextual understanding [16]. This capacity to evaluate words in their contextual environment enhances its handling of words with multiple meanings.
- GloVe creates a static embedding for each word by analyzing its co-occurrence within a large corpus, but it does not take into account the word's context within a sentence. Since textual content in video models is frequently dynamic and context-sensitive, GloVe is unable to grasp the word meanings based on their specific usage resulting in a less precise interpretation and processing of text in videos [14].

Hence, it appears that the advanced contextual comprehension and flexibility of BERT make it more suitable for training video models such as VSLNet. This is due to its superior capability to handle the complex textual elements asso-

ciated with videos, compared to the static and less context-aware GloVe embeddings [16].

4.3. Experiments on Natural Language Answers

After implementing our extension, we conducted an analysis using specific metrics to evaluate the performance of our extension. Following are the results achieved:

Metric	Score
Average `METEOR`	0.5612
Average `ROUGE-L precision`	0.5724

`METEOR` Score (0.5612) suggests that the translations or summaries produced by our model are optimally aligned with the ground truths in terms of both accuracy and order. `ROUGE-L Precision` Score (0.5724) shows that more than half of the longest common subsequences (LCS) in the model output were also present in the ground truth, confirming that the predicted content is relevant and mostly accurate.

In brief, both scores indicate that the model is highly competent at understanding and generating text that matches the ground truth. They reflect a balance between completeness (capturing all pertinent content) and precision (avoiding the inclusion of irrelevant content).

To analyze some qualitative results, Figure 3 shows the queries, the answers generated by LLaVA, and three frames extracted from specific points within the prediction interval. By manually testing the visual representation, we found that 10%, 50%, and 90% of the video's duration are good points for extracting qualitative results, since they display useful information about the segment.

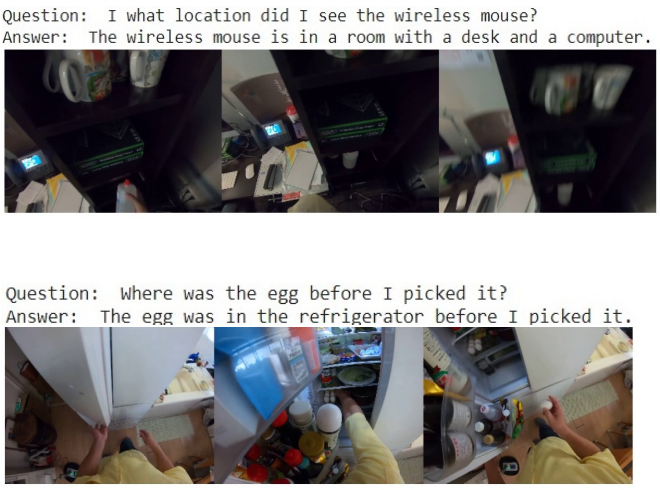

Figure 3. Qualitative results for natural language response generation through Video-LLaVA

5. Conclusion

This paper leverages the comprehensive Ego4D dataset, featuring egocentric videos, to advance research in first-person perspective. It concentrates on the Ego4D Episodic Memory benchmark, which focuses on responding to questions about specific past events depicted in the videos, with an emphasis on refining Natural Language Queries and employing the Video-LLaVA model, the project significantly extends the interpretability of the obtained results by generating natural language responses to queries. Challenges such as computational constraints and difficulties in processing videos in low-light conditions were encountered, suggesting that preprocessing could enhance results. These will be thoroughly explored as avenues for future work. Overall, our work furthers the functionality and practicality of egocentric video datasets and the analysis of first-person video data.

6. Project Source Code

The project codes can be found in the following public GitHub Repository: ***NLQ in Egocentric Videos***.

References

[1] Leonard Bärmann and Alex Waibel. Where did i leave my keys?-episodic-memory-based question answering on egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1560–1568, 2022.

[2] Rohan Chandra, Mridul Mahajan, Rahul Kala, Rishitha Palugulla, Chandrababu Naidu, Alok Jain, and Dinesh Manocha. Meteor: A massive dense & heterogeneous behavior dataset for autonomous driving. *arXiv preprint arXiv:2109.07648*, 1(2), 2021.

[3] Yisen Feng, Haoyu Zhang, Yuquan Xie, Zaijing Li, Meng Liu, and Liqiang Nie. Objectnlq@ ego4d episodic memory challenge 2024. *arXiv preprint arXiv:2406.15778*, 2024.

[4] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens Van Der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16102–16112, 2022.

[5] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[6] Rishi Hazra, Brian Chen, Akshara Rai, Nitin Kamra, and Ruta Desai. Egotv: Egocentric task verification from natural language task descriptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15417–15429, 2023.

[7] Zhijian Hou, Lei Ji, Difei Gao, Wanjun Zhong, Kun Yan, Chao Li, Wing-Kwong Chan, Chong-Wah Ngo, Nan Duan,

and Mike Zheng Shou. Groundnlq@ ego4d natural language queries challenge 2023. *arXiv preprint arXiv:2306.15255*, 2023.

[8] Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. Egotaskqa: Understanding human tasks in egocentric videos. *Advances in Neural Information Processing Systems*, 35:3343–3360, 2022.

[9] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[10] Jun-Ping Ng and Viktoria Abrecht. Better summarization evaluation with word embeddings for rouge. *arXiv preprint arXiv:1508.06034*, 2015.

[11] Thi-Hoa-Cuc Nguyen, Jean-Christophe Nebel, and Francisco Florez-Revuelta. Recognition of activities of daily living with egocentric vision: A review. *Sensors*, 16(1):72, 2016.

[12] Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. Rank & sort loss for object detection and instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3009–3018, 2021.

[13] Himanshu Singh and Himanshu Singh. Data processing in aws. *Practical Machine Learning with AWS: Process, Build, Deploy, and Productionize Your Models Using AWS*, pages 89–117, 2021.

[14] Hao Zhang. Towards temporal sentence grounding in videos. 2022.

[15] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Natural language video localization: A revisit in span-based question answering framework. *IEEE transactions on pattern analysis and machine intelligence*, 44(8):4252–4266, 2021.

[16] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. *arXiv preprint arXiv:2004.13931*, 2020.

[17] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597, 2023.