From Egocentric Videos to Textual Answers: A Two-Stage Pipeline for NLQ-Based Video QA

Giovanni Tagliaferri, Marco Del Core, Davide Jannussi
Github: https://github.com/MarcoDelCore/ego-aml24

Abstract

The Natural Language Query (NLQ) task focuses on identifying temporal segments within a video that correspond to a given query. This project introduces a pipeline that integrates the NLQ task, leveraging VSLNet and VSLBase models, with VideoQA using Video-LLaVA, applied to egocentric videos from the Ego4D dataset. By first isolating relevant video clips through NLQ, the pipeline enables VideoQA models to generate textual answers, overcoming the challenges posed by the length and unstructured nature of egocentric videos. This approach combines the strengths of NLQ for precise temporal localization and VideoQA for producing human-like responses, facilitating intuitive and accurate query-based video understanding.

1. Introduction

The Natural Language Queries (NLQ) task defined in Ego4D benchmark [1] is defined as an extraction of a temporal segment in which, given a video clip and a query expressed in natural language like “Where is object X?” or “Who did person Z interact with?”, the model is asked to predict the temporal segment where the answer is visible or deducible. Unlike generating textual answers, NLQ extracts video clips corresponding to the query, leaving interpretation to the observer. This is particularly useful for processing long and unstructured egocentric videos, which are computationally challenging for Video Language Models (VLMs). A potential extension involves using the identified intervals as input for VideoQA models to generate human-like textual descriptions.

The task utilizes the Ego4D dataset [1], a large-scale egocentric video collection encompassing 3,670 hours of diverse activities. We leverage pre-extracted features from Omnivore [3] and EgoVLP [2], both utilizing temporal windows of 16 frames with a stride of 16. The project initially focuses on NLQ, employing span-based question-answering approaches implemented via VSLNet and VSLBase models [5]. These models, trained on these feature sets, are compared against Ego4D’s official baseline results, which use SlowFast features [4].

Additionally, two variations of the VSLNet and VSLBase models are explored to identify potential improvements: the first is replacing the original text encoder (BERT [6]) with GloVe [7], a pre-trained word embedding model, to see how different text representations influence the models’ ability to interpret queries; the second is using separated encoders for text and video to better capture the distinct characteristics of each modality, potentially enhancing the alignment between video and textual information.

After completing the NLQ task, the focus shifts to generating textual descriptions for the extracted video clips using Video-LLaVA (Large Language and Vision Assistant) [8], a VideoQA model designed to generate textual answers from a video clip and its corresponding query. Once the NLQ task predictions are compared with the Ego4D baseline results, 50 queries are manually selected and annotated with the ground truth. The related video clips are downloaded and provided to Video-LLaVA with the queries as inputs. Its predictions are then compared to the ground truth and evaluated with different metrics.

2. Related works

2.1. NLQ Task

As briefly explained in 1, the NLQ task aims to retrieve a temporal segment inside an untrimmed video that answers to a given language query. Similarly to the span-based question answering (QA) task in natural language learning where the purpose is to extract a span of words from a text related to a give query, the idea here is to consider a video as a text-passage and extract a temporal span (so a time interval) where the query is answered. This span-based approach has two problems: first, in a video the events are continuous and changes between frames are usually very small, while natural language is inconsecutive and words have a syntactic structure and different meanings; second, humans are less sensitive to small shifts between video frames, as they do not significantly affect the understanding of the content. In contrast, in natural language, even a small change between words can alter the meaning of a sentence, making language more sensitive to minor variations.

VSLBase [5] is a baseline model that applies a span-based QA approach to the NLVL task (Natural Language Video Localization), treating visual features as text passages and the target moment as an answer span. However, it does not address key challenges between video and language. VSLNet improves on this by introducing a Query-Guided Highlighting (QGH) strategy, where the target moment and its context are treated as the foreground, and the rest as the background. This highlights a larger region, providing better context and allowing the model to focus on subtle differences between frames, thus improving localization accuracy.

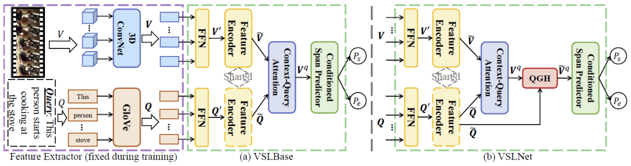


Figure 1. VSLBase and VSLNet architectures taken from [5].

2.1.1 VSLBase

VSLBase is a span-based question-answering (QA) model developed for the Natural Language Video Localization (NLVL) task. It processes visual features extracted from videos and textual word embeddings by projecting both modalities into a common dimensional space.

The model employs a shared feature encoder, consisting of convolutional layers and a multi-head attention mechanism to generate enriched representations of both visual and textual inputs. A context-query attention (CQA) mechanism is utilized to capture interactions between the video content and the language query, enabling effective alignment and localization of relevant video segments.

For predicting the temporal boundaries of the target segment, VSLBase uses two stacked unidirectional LSTM networks that sequentially determine the start and end points based on the encoded features. However, VSLBase does not fully address the challenges of aligning continuous video events with the discontinuous structure of natural language, highlighting the need for further advancements addressed in subsequent models like VSLNet.

2.1.2 VSLNet

VSLNet builds upon the VSLBase model by introducing a Query-Guided Highlighting (QGH) strategy to better align video and text modalities. The QGH mechanism distinguishes the target moment, which corresponds to the language query, from the surrounding background content. This is achieved through a binary classification module that assigns relevance scores to each visual feature, effectively highlighting the most pertinent regions of the video. By focusing on these highlighted segments, VSLNet enhances the model's ability to accurately localize the target temporal segment, addressing the limitations of VSLBase in handling the nuanced differences between continuous video events and discontinuous natural language queries. This improvement leads to more precise and reliable localization performance.

2.2. Video Question Answering

Video Question Answering (Video QA) is a task where models analyze videos and answer questions about their content by understanding both visual and textual information. Traditional methods often handle video and text features separately, combining them later, which can cause misalignment and reduce effectiveness.

Large Vision-Language Models (LVLMs) are designed to address these challenges by creating a unified representation of visual and textual data. For example, Video-LLaVA aligns video features with language features, enabling better integration and understanding of multi-modal information. This approach improves performance on many benchmarks, showing how LVLMs can effectively handle both images and videos for tasks like Video QA.

2.2.1 Video LLaVa

The Video-LLaVA model is structured to unify visual and textual modalities for enhanced performance in multimodal tasks. The model consists of several key components: LanguageBind encoders f_V to extract features from raw visual data (images or videos), a large language model (LLM) f_L, visual projection layers f_P, and a word embedding layer f_T. The word embedding layer converts textual input, such as queries or questions, into dense vector representations. These word embeddings allow the model to process textual information in a way that aligns with the visual features extracted from the video and image encoders. Once the visual features are encoded into a unified representation, they are combined with the word embeddings and passed into the LLM for generating responses. This architecture enables the model to learn from a shared feature space, facilitating interactions between text and images or videos, and allowing the model to better understand and generate contextually relevant responses based on multimodal inputs.

3. Methods

The primary objective of this work is to provide accurate textual answers given a natural language query and a corresponding egocentric video. To accomplish this, we designed a two-stage pipeline.

The first stage focuses on identifying the video segment relevant to the query. To achieve this, we used two models, VSLNet and VSLBase [5], both trained to predict the start and end timestamps, ensuring that only the portion of the video likely to contain the answer is selected.

The second stage generates a textual response by processing the selected video segment and the input query using Video-LLaVA [8]. This approach minimized unnecessary computations and enhanced the accuracy of the answers.

3.1. Dataset and Feature Usage

In this project, we used the Ego4D dataset [1], focusing on the Natural Language Query (NLQ) task. The ”annotations” subset was utilized for training and evaluation, providing temporal labels that indicate the start and end times of relevant video segments.

For feature representation, we relied on the pre-extracted EgoVLP [2] and Omnivore [3] features from Ego4D. Each type of feature was used independently to train the two models, resulting in four different model-feature combinations. The goal of this approach was to compare the performance of the models using these distinct feature sets and identify the best-performing configuration.

3.2. Model Training and Evaluation Process

For each combination of model (VSLBase or VSLNet) and feature type (EgoVLP or Omnivore), we conducted a structured process involving separate phases of training, validation, and testing. During the training phase, the annotated segments from the dataset were used to help the models learn to predict the correct start and end times for relevant video segments corresponding to specific natural language queries.

Our primary objective was to fine-tune the models to achieve accurate temporal localization, ensuring they could reliably identify the video portions that best matched the input queries. This rigorous approach allowed us to compare the effectiveness of different feature sets and models, ultimately selecting the combination with the best performance for integration into our final pipeline.

3.3. Model Variations

To further analyze the performance of VSLNet, we implemented and tested two architectural variations. For both variations, we repeated the training, testing, and validation processes using the Omnivore features. The performance of these modified models was then compared against the results obtained with VSLNet.

3.3.1 Replacing the Text Encoder

In the first variation, we replaced the original BERT-based text encoder [6] with a GloVe-based encoder [7]. This modification aimed to evaluate how a simpler, pre-trained embedding model like GloVe would impact the overall performance of VSLNet compared to the more complex BERT encoder.

3.3.2 Using Non-Shared Encoders for Video and Text

In the second variation, we used separate encoders for the video and text modalities instead of sharing a single encoder. The motivation behind this approach was to allow each modality to be processed independently, potentially leading to better feature extraction tailored to the specific nature of video and text data.

3.4. Selection of NLQ Queries for Ground Truth Annotation

To advance in our pipeline, we selected 50 natural language queries (NLQs) for which VSLNet, during its final training epoch with Omnivore features, had accurately predicted the correct video segments. Specifically, we focused on those predictions that showed a high degree of overlap with the annotated ground truth. A prediction was considered accurate if the Intersection over Union (IoU) between the predicted time interval and the ground truth interval exceeded 0.8.

The selection process ensured that each video was associated with only one query, thus avoiding redundancy and ensuring a variety of video segments for further analysis. After selecting the queries, we manually annotated the corresponding textual ground truth for each query, creating a reliable reference for the next stages of our research. This carefully curated set of data was essential for the subsequent task in the pipeline: generating textual answers based on the selected video segments.

3.5. Video Extraction for Selected Segments

After selecting the 50 NLQ queries and their corresponding video segments, the relevant video files were retrieved individually. Using the timestamps derived from the predicted intervals, we applied ffmpeg to extract the specific segments of interest from each video. These extracted segments were then provided as input to Video-LLaVa for further analysis. By focusing exclusively on the relevant portions of the videos, we ensured that the Video Language Model (VLM) received the precise content necessary for answering the queries, optimizing the model's performance in the subsequent stages of the pipeline.

3.6. Adopting a Video Question Answering Model

In the final stage of our pipeline, we integrated a video question answering (VideoQA) model to generate textual responses based on the previously identified video segments and their corresponding queries. Specifically, we employed Video-LLaVA, a model designed to interpret visual content and answer natural language queries. Each query, along with its associated video clip, was provided as input to the model, which then produced a direct textual answer.

By supplying only the most relevant video segments—those precisely localized by VSLNet in earlier steps—we aimed to present the model with a well-defined context that directly pertained to the given query. This design choice ensured that the model could focus exclusively on the key information needed to produce accurate answers, thereby reducing the risk of irrelevant or erroneous outputs.

Moreover, this selective input approach allowed us to optimize the computational efficiency of the VideoQA process. Instead of processing entire videos, which would introduce unnecessary complexity and dilute the contextual relevance of the input, the model worked with concise video snippets, ensuring that the visual data was directly aligned with the query's intent. This alignment was critical in achieving high-quality responses, as it minimized noise and emphasized the salient visual cues required for accurate answer generation.

To validate the performance of Video-LLaVA in this setup, we systematically recorded its responses and later compared them to the manually curated ground truth answers. This comparison provided an objective measure of the model's accuracy in understanding and responding to the selected video-query pairs, forming the basis for a detailed evaluation of the overall effectiveness of our proposed pipeline.

4. Experiments

This section presents the results, compares the models, and analyzes their performance. The experiments are divided into two main parts: the evaluation of the NLQ task on the Ego4D dataset and the extension to video question answering.

4.1. Natural Language Queries (NLQ)

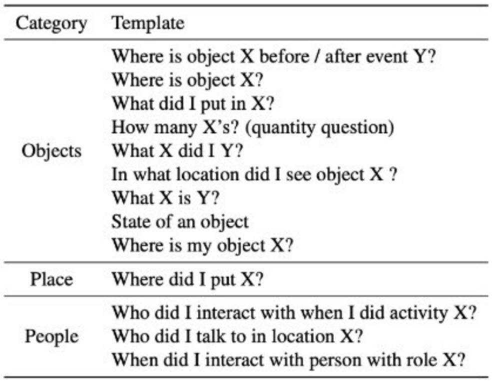
The first part of the project focuses on the Natural Language Queries task, where the model predicts the time interval in which the answer to a natural language query is visible or deducible in the video. The dataset used, along with the results, is detailed below.

4.1.1 4.1.1 Dataset

Dataset: Ego4D [1] - Natural Language Queries (NLQ) The Ego4D NLQ benchmark focuses on temporal segment prediction, where natural language queries are linked to specific video segments.

Key Characteristics

- **Annotated queries:** Approximately 19,000.
- **Total video duration:** Around 227 hours.
- **Annotations:** Provided in JSON format (`nlq_{split}.json`), with queries and their corresponding temporal segments.
- **Query types:** Derived from 13 predefined templates, covering objects, places, and people (Figure 2).



Category	Template
Objects	Where is object X before / after event Y?
	Where is object X?
	What did I put in X?
	How many X's? (quantity question)
	What X did I Y?
	In what location did I see object X ?
	What X is Y?
	State of an object
	Where is my object X?
Place	Where did I put X?
People	Who did I interact with when I did activity X?
	Who did I talk to in location X?
	When did I interact with person with role X?

Figure 2. Query templates from the Ego4D NLQ benchmark.

Statistical Insights To provide a deeper understanding of the dataset, we include the following statistical analyses:

- **Query Distribution Among Templates:** Queries are distributed across the 13 predefined templates. Figure 3 highlights the frequency of queries within each template.
- **Scenario Distribution:** Queries span various scenarios like cooking, cleaning, eating, and more. Figure 4 shows cooking and cleaning are the most frequent by far.
- **Clip and Segment Duration:** The majority of video clips are approximately from 450 to 500 seconds long, with a smaller subset extending up to 1200 seconds. Temporal segments linked to queries are generally concise, with durations typically under 25 seconds. Figures 5 and 6 illustrate these distributions, highlighting the variability in clip lengths and the brevity of the corresponding query/answer segments.

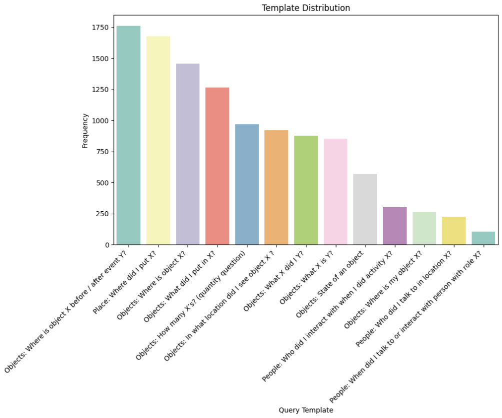


Figure 3. Distribution of queries across the 13 templates. This highlights the diversity and frequency of query types in the dataset.

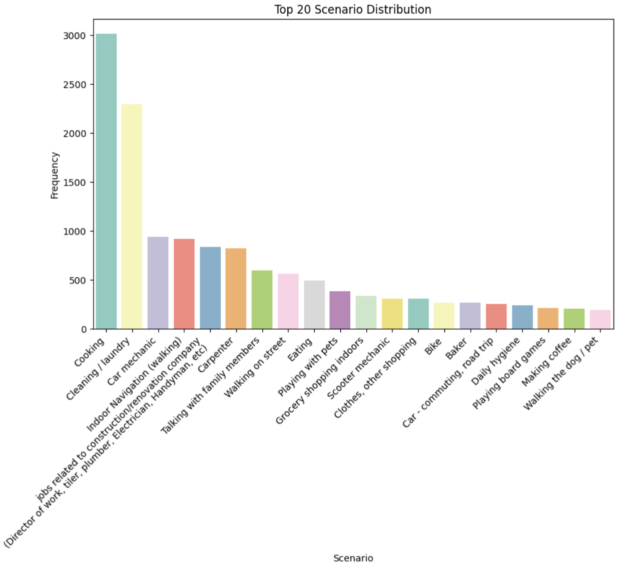


Figure 4. Scenario Distribution. Histogram showing the frequency of queries across top 20 scenarios.

4.1.2 Results on Ego4D NLQ Benchmark

Metrics Used To evaluate the models' performance, the following metrics were utilized:

- **mIoU (Mean Intersection over Union):** This metric evaluates the overlap between the predicted and ground truth segments. It provides a measure of temporal accuracy by calculating the average IoU across all predictions.

- **Rank@1, Rank@3, Rank@5:** These metrics measure the percentage of queries for which the correct temporal segment is within the top 1, 3, or 5 predicted segments. A segment is considered correct if it achieves a Mean Intersection over Union (mIoU) greater than or equal to the specified threshold (e.g., 0.3 or 0.5).

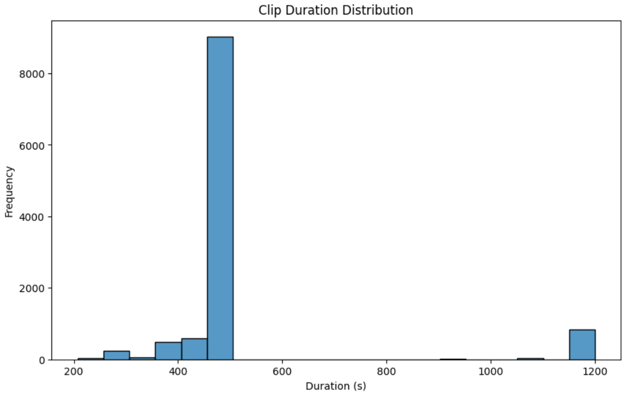


Figure 5. Clip Duration Distribution. The histogram shows the majority of clips are around 400 seconds long, with a smaller subset extending up to 1200 seconds.

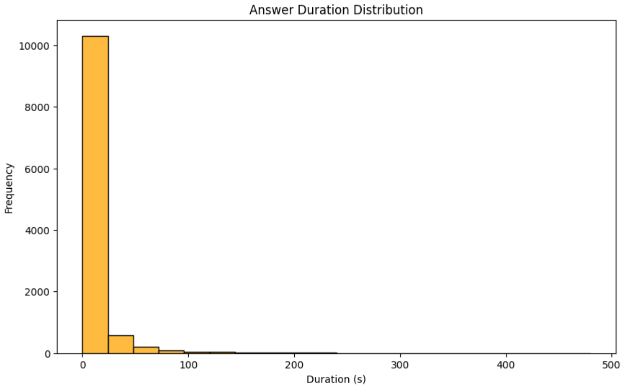


Figure 6. Answer Duration Distribution. Temporal segments linked to queries are typically under 20 seconds.

Results Overview The results of the evaluation are presented in Table 1 and Table 2. Table 1 compares the performance of **VSLNet** and **VSLBase** using features from **Omnivore** and **EgoVLP** to the Ego4D NLQ baseline trained and evaluated on **SlowFast** features [4] (first two rows). Table 2 provides a comparison of VSLNet with Omnivore features against variations of the architecture. Specifically, it includes results obtained by:

- Replacing the **BERT**-based text encoder with **GloVe**.

- Using **non-shared encoders** for video and text modalities, allowing each modality to be processed independently for potentially better feature extraction.

This comparison highlights the impact of these architectural modifications on the overall performance.
Key Observations:

- Both **Omnivore** and **EgoVLP** features consistently outperform **SlowFast** features across all metrics considered (note that Rank@3 metrics are not provided for the baseline and are therefore missing in the first two rows).

- The **VSLNet** model outperforms **VSLBase** in all the metrics, with both Omnivore and EgoVLP features, this results demonstrates the effectiveness of Query-Guided Highlighting (QGH) introduced in VSLNet.

- For **VSLBase**, some metrics show better performance with **Omnivore** features, while others favor **EgoVLP**. In contrast, **VSLNet** utilizing **EgoVLP** features achieve higher scores compared to the same models trained with **Omnivore** features except for Rank@1 with mIoU@0.3 where the model trained on Omnivore is slightly better. This suggests that **EgoVLP** features are well-suited for egocentric tasks due to their alignment with domain-specific characteristics.

- Replacing the **BERT**-based encoder with **GloVe** leads to a significant drop in performance.

- Using **non-shared encoders** for video and text modalities results in slightly lower performance compared to the standard VSLNet.

Overall, the best-performing configuration appears to be **VSLNet** with **EgoVLP** features, which are specifically designed for egocentric videos. This highlights the effectiveness of both the **EgoVLP** features for egocentric tasks and the **Query-Guided Highlighting (QGH)** mechanism in **VSLNet**.

4.2. Extension Video Question Answering

In this part, we extended the pipeline to incorporate a Video Question Answering (VideoQA) task. For this purpose, we selected 50 video clips from the correct predictions made by the previous **VSLNet** model.

A prediction was defined as correct if at least one of the top five video segments returned by **VSLNet** overlapped temporally with the ground truth by at least 80% IoU. Among these correct predictions, we chose a diverse subset of clips where the model had successfully localized relevant video segments, ensuring high-quality input for the VideoQA task.

4.2.1 Results on Extension with LLaVA (VideoQA)

By manually evaluating the predictions, we found that 18 out of 50 queries provided factually correct responses. However, a manual approach is impractical when scaling to a larger number of queries. This highlights the necessity of robust evaluation metrics to systematically assess the model's performance on a broader scale.

Metrics Used To evaluate the performance of the Video Question Answering (VideoQA) task, we utilized the following metrics:

- **BLEU [11]:** A metric for evaluating the quality of the generated text by comparing it with the ground truth based on n-gram overlaps.

- **ROUGE-L [9]:** A metric that evaluates the overlap of the longest common subsequence (LCS) between the generated text and the reference.

- **BERTScore (Precision, Recall) [10] :** Evaluates the semantic similarity between the generated text and the ground truth by leveraging embeddings from pre-trained BERT models:
  - **Precision:** Measures how much of the generated text aligns semantically with the ground truth.
  - **Recall:** Measures how much of the ground truth is captured by the generated text.

Below, we present the results of the VideoQA task using the above metrics across 50 queries. Each graph highlights the performance of the model for a specific metric.

4.2.2 Qualitative Results

Below, we present a selection of qualitative results to illustrate the model's performance in cases where it provided good results. These examples provide context for the subsequent metric analysis, highlighting scenarios where the model delivered accurate and well-aligned predictions.

Example 1

Query:	Did I leave the fridge open?
Prediction:	Yes, you left the fridge open.
Ground Truth:	Yes, I left the fridge open.
Metrics:	
Precision:	0.976
Recall:	0.976
BLEU:	0.594
ROUGE-L:	0.833

Here, the model performed strongly across all metrics, producing a prediction that was both semantically accurate and factually aligned with the ground truth.

Model	**Rank@1** IoU@0.3	**Rank@1** IoU@0.5	**Rank@3** IoU@0.3	**Rank@3** IoU@0.5	**Rank@5** IoU@0.3	**Rank@5** IoU@0.5	**mIoU**
2D-TAN (SlowFast)	5.04	2.02	-	-	12.89	5.88	-
VSLNet (SlowFast)	5.45	3.12	-	-	10.74	6.63	-
VSLNet (Omnivore)	**6.43**	3.64	10.66	6.69	12.96	8.18	5.01
VSLBase (Omnivore)	5.96	3.48	9.94	6.17	12.67	8.00	4.71
VSLNet (EgoVLP)	6.40	**4.03**	**11.15**	**7.38**	**13.84**	**9.22**	**5.18**
VSLBase (EgoVLP)	5.70	3.38	10.22	6.58	12.52	7.92	4.78

Table 1. Performance on the Ego4D NLQ Validation Set (Val) using different models and feature types. The highest values for each metric are in bold.

Model	**Rank@1** IoU@0.3	**Rank@1** IoU@0.5	**Rank@3** IoU@0.3	**Rank@3** IoU@0.5	**Rank@5** IoU@0.3	**Rank@5** IoU@0.5	**mIoU**
VSLNet (Omnivore)	6.43	3.64	10.66	6.69	12.96	8.18	5.01
GloVe	4.21	2.27	8.21	4.49	10.79	6.09	3.52
Encoders	6.20	3.67	10.43	6.40	12.21	8.05	4.94

Table 2. Comparison of different feature encoders for VSLNet on the Ego4D NLQ benchmark.

Example 2

Query:	In what location did I last see the car?
Prediction:	I last saw the car on a dirt road.
Ground Truth:	On a dirt road near the tent.
Metrics:	
Precision:	0.889
Recall:	0.908
BLEU:	0.168
ROUGE-L:	0.5

The prediction captures most of the ground truth, leading to a high ROUGE-L score. However, BLEU is lower because it evaluates how correct the prediction is, penalizing additional content like *"last saw"* in the prediction, which is not present in the ground truth, despite overall semantic alignment.

Example 3

Query:	Who did I talk to at the ATM place?
Prediction:	The woman who is standing in front of the ATM machine.
Ground Truth:	I talked to a woman.
Metrics:	
Precision (BERTScore):	0.860
Recall (BERTScore):	0.902
BLEU:	0.040
ROUGE-L:	0.125

Despite the semantic similarity between the prediction and ground truth, BLEU and ROUGE scores are low due to phrasing differences and lack of exact n-gram matches. However, BERTScore Precision and Recall capture the semantic alignment, showing their strength in evaluating meaning over lexical overlap.

4.2.3 Quantitative Results

BLEU We observe that the queries receive a low score for this metric, primarily due to the limitations of BLEU. BLEU does not account for semantic meaning, meaning synonyms or paraphrased content conveying the same idea may still result in a low score. Additionally, it penalizes short responses, even when they are correct. This is especially problematic for concise but accurate answers.

Example

Query:	Did I wash the green pepper?
Prediction:	Yes, you washed the green pepper in the sink.
Ground Truth:	Yes, you did it.
BLEU:	0.0446

ROUGE ROUGE prioritizes recall, which measures how much of the ground truth is captured by the generated text. This metric has shown better performance than BLEU in our VideoQA task because it is more forgiving when synonyms or paraphrasing occur, as long as the key information

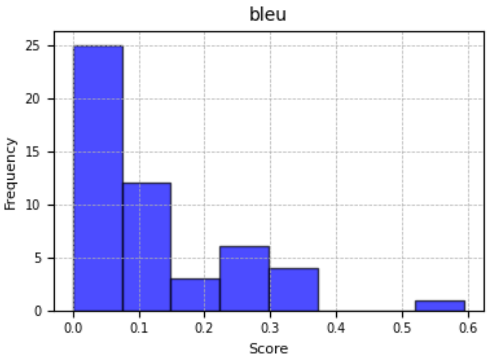


Figure 7. Histogram of BLEU scores across 50 queries.

is present. However, it struggles with fully addressing semantic variations, especially in cases where the prediction conveys the correct meaning but uses different phrasing or sentence structures (Example 3).

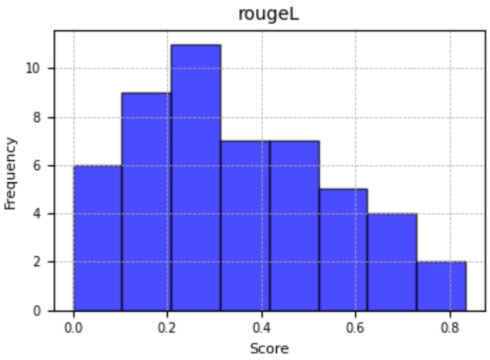


Figure 8. Histogram of ROUGE-L scores across 50 queries.

BERTScore To address the limitations of BLEU and ROUGE with synonyms and rephrased content, we use BERTScore Precision and Recall, which rely on contextual embeddings rather than exact matches. Recall is particularly important as it ensures that most of the information in the ground truth is captured in the generated response. While the results appear significantly better with this metric, BERTScore can assign high scores to predictions that are semantically similar to the ground truth but factually incorrect or contextually irrelevant.

Example

Query:	In what room did I see the tissue paper?
Prediction:	The tissue paper was seen in a bathroom.
Ground Truth:	You saw the tissue paper in the kitchen.
Metrics:	
Precision:	0.927
Recall:	0.932

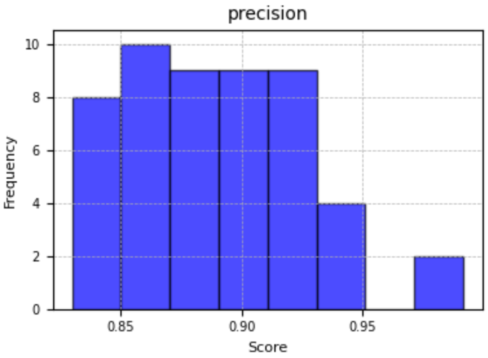


Figure 9. Histogram of BERTScore Precision scores across 50 queries.

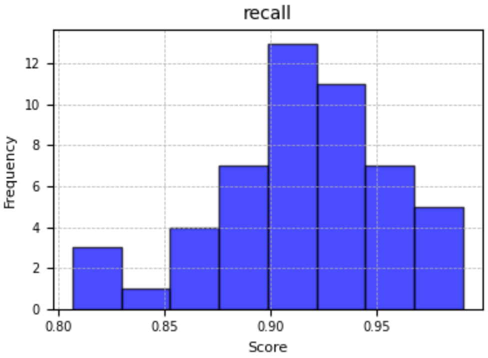


Figure 10. Histogram of BERTScore Recall scores across 50 queries.

4.2.4 Addressing Metric Limitations

To address the limitations of existing metrics, a potential solution could involve defining a taxonomy of possible responses, relying on a limited set of clear and predefined words for each response, and evaluating how effectively the ground truth is captured in terms of recall.

5. Conclusion

This work proposed a two-step approach to generate textual answers from natural language queries in egocentric videos. By combining NLQ models (VSLBase and VSLNet) for segment retrieval with VideoQA models for answer generation, we provided a solution for handling long and unstructured videos. Our results showed that this method effectively bridges video segment identification and textual response generation while reducing computational overhead.

References

[1] K. Grauman *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 3, 4

[2] K. Q. Lin *et al.*, "Egocentric video-language pretraining," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7575–7586, 2022. 1, 3

[3] R. Girdhar, M. Singh, N. Ravi, L. Van Der Maaten, A. Joulin, and I. Misra, "Omnivore: A single model for many visual modalities," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16102–16112. 1, 3

[4] C. Feichtenhofer *et al.*, "Slowfast networks for video recognition," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 1, 5

[5] H. Zhang *et al.*, "Span-based Localizing Network for Natural Language Video Localization," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. 1, 2, 3

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019. 1, 3

[7] J. Pennington, R. Socher, e C. D. Manning, "GloVe: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 1, 3

[8] B. Lin, B. Zhu, Y. Ye, M. Ning, P. Jin, and L. Yuan, "Video-llava: Learning united visual representation by alignment before projection," *arXiv preprint arXiv:2311.10122*, 2023. 1, 3

[9] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. 6

[10] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004. 6

[11] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: A method for automatic evaluation of machine translation," *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002. 6