LLM-Based Augmentation for NLQ in Ego4D

Andrea Cucchietti
Politecnico di Torino, Italy
s[redacted]@studenti.polito.it

Annalisa Belloni
Politecnico di Torino, Italy
s[redacted]@studenti.polito.it

Davide Elio Stefano Demicheli
Politecnico di Torino, Italy
s[redacted]@studenti.polito.it

Abstract

This work aims to delve into the domain of egocentric vision and its integration with Natural Language, in order to achieve video understanding. In particular, we focus our attention on the NLQ task of the Episodic Memory benchmark proposed by Ego4D, leveraging the VSLBase and VSLNet models. Various configurations of these two models are tested, analyzing the effects of different choices of vision and text encoders, and the respective outcomes are presented in detail. Furthermore, we propose to pre-train the best performing configuration on an LLM-generated dataset of queries. This approach leads to better performance after fine-tuning on the original benchmark data.

1. Introduction

Egocentric Vision captures human interactions from a unique perspective, using cameras mounted directly on the user's head as they perform various actions. The release of large-scale datasets has advanced the frontier of first-person perception, encouraging the research community to explore new methods for learning from egocentric videos, such as using natural language queries for video understanding.
A big step forward was taken with the introduction of the Ego4D dataset: the world's largest first-person video ML dataset and benchmark suite [4]. The dataset contains 3,670 hours of densely narrated daily-life activity videos and a great number of annotations associated with five benchmark tasks.

Among these, we focused on the Episodic Memory benchmark, which aims to make past first-person visual experiences queryable. To be precise, given an egocentric video and a query, the Ego4D Episodic Memory task consists in pinpointing where the answer can be found within a long video.

In particular, the queries we address are natural language queries (NLQ), meaning each query is expressed in text (e.g., "What did I put in the drawer?"), and the relative output response is the temporal window where the answer is inferable, capturing the essence of Natural Language Video Localization (NLVL). Being a vision-language understanding task, NLVL requires both computer vision and natural language processing techniques, leading us to adopt a cross-modal reasoning approach.

Figure 1. An example of NLQ task with query and relative temporal response window.

Inspired by the improvements of visual-textual embeddings and query-localization models, we tried to face one of the most relevant issues of the NLQ task: the limited number and scope of annotations, which is reflected in difficulties for the existing methods in learning about concepts that are poorly represented in training, for instance when we have complex queries addressing interactions between several visual entities. This restricted quantity of annotations is due to the significant efforts that they require to be created. However, the dataset also includes narrations, another useful source of textual information for the videos, which are not only considerably easier to annotate but also more varied, significantly more numerous, and can be adapted to be employed for the NLQ task.

Specifically, we propose leveraging timestamped narrations to pre-train query-localization models. By utilizing a large language model (LLM) to automatically generate queries from these narrations, we aim to augment the existing dataset, providing more varied examples of queries, thereby improving the performance of query-localization models. This approach not only capitalizes on the abundance of narration data but also reduces the manual effort involved in query annotation, paving the way for more efficient and scalable data augmentation methods in wearable camera video analysis.

2. Related Work

The field of egocentric vision has seen significant advancements over the past few years. In this section, we review the most relevant literature, focusing our attention on the NLQ task and briefly outlining the context, the existing approaches, and the building blocks on which our work is based.

2.1. Natural language video localization

NLQ is a video grounding task in the Ego4D episodic memory benchmark which consists of temporally localizing where the answer of a natural language query can be seen in an egocentric video. Already existing video grounding methods like 2D-TAN [10] and VSLNet [9] have subsequently been adapted to handle this new task.

In particular, 2D Temporal Adjacent Network (2D-TAN) is a single-shot framework for moment localization, able to encode the adjacent temporal relations while learning discriminative features for matching video moments with the corresponding expressions. Its core idea is to retrieve a moment on a two-dimensional temporal map, which considers adjacent moment candidates as the temporal context.

VSLNet, instead, solves the NLVL task with a multimodal span-based QA framework, employing a query guided highlighting (QGH) strategy which highlights the region where the model has to search.

Our objective, similarly to what has been done by NaQ [7], is to further enhance these methods by employing data augmentation with narrations-based queries. The densely annotated narrations available in the EGO4D dataset are exploited as an additional and more general data source to improve the model's ability to perform the NLQ task. This is realized by adapting and converting the narrations and their timestamps into NLQ annotations. However, while NaQ employs the same narrations as queries (exploiting encoders that can effectively adapt to the difference between declarative sentences and questions), we instructed an LLM to generate the queries for the dataset augmentation.

2.2. Text Encoders

There are several approaches to the problem of text encoding. Two relevant examples are BERT [1] and GloVe [6]. These two language embedding models both fulfill the function of transforming textual input into vector representations. However, they present significantly different structures, which affect their characteristics and performance. The major difference between the two models is that GloVe is a static word embedding model: each word has a single representation vector that does not change, regardless of the context in which the word appears. In contrast, BERT is a contextual embedding model with a representation of each word that is dynamic and changes depending on the context.

The BERT embedding model learns word representations based on both preceding and following contexts (bidirectional) and is based on the Transformer architecture.
Given these two different natures, the GloVe-based text encoder is typically simpler and less computationally intensive, whereas the BERT-based encoder is more complex, requiring greater computational power, but also more powerful in the understanding.
We separately employed these two types of encoders within the VSLNet model to test and compare the results obtained with each approach.

2.3. Vision Encoders

Training models end-to-end on video datasets is often impractical due to the large computational budgets and time required. A common strategy adopted to bypass this problem is to leverage pre-trained models and fine-tune them for the specific downstream task we need to solve. Among these pre-trained models, we focused our attention on Omnivore [3] and EgoVLP [5].

Omnivore is a vision transformer model designed to handle multiple types of visual inputs such as images, videos, and 3D data with a single model architecture. The key idea behind Omnivore is to unify these different types of visual data representations into a common framework, allowing the final embeddings to be used for a wide variety of vision tasks. It utilizes a vision transformer architecture, which has proven effective in capturing long-range dependencies and contextual information in visual data. Pre-training on large, diverse datasets allows Omnivore to learn robust and general features that can be transferred to specific tasks, as in our case.

EgoVLP, instead, is a framework for video-language pretraining specifically tailored for egocentric video data. It addresses the challenges posed by unstructured and noisy data in the Ego4D dataset by curating a pre-training dataset called EgoClip. Furthermore, the model's ability to align video and textual representations is obtained using a contrastive learning approach exploiting the weak supervision of free form textual narrations.

2.4. LLM

Gemma-2b-it-V2 [8] is part of a family of lightweight, state-of-the-art open models released by Google in 2024. The model has already been trained on a large dataset of textual data and serves as a valid and versatile tool for a wide range of natural language processing tasks. Further specialization in a particular task is possible by performing fine-tuning on that specific task. Specifically, we used the *Gemma-2b-it-V2* model, since it provides results of acceptable quality without requiring considerable computational resources.

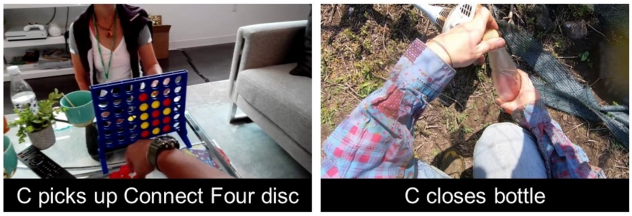

Figure 2. Example of narrations. The letter C indicates the camera wearer.

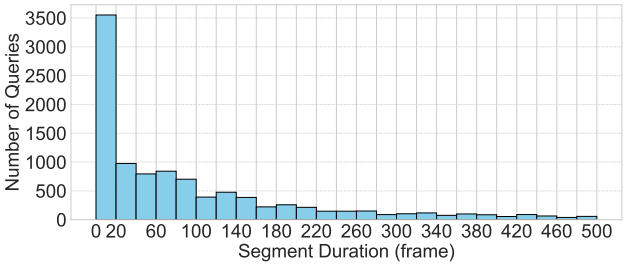

Figure 3. Number of Queries by Response-Segment Duration.

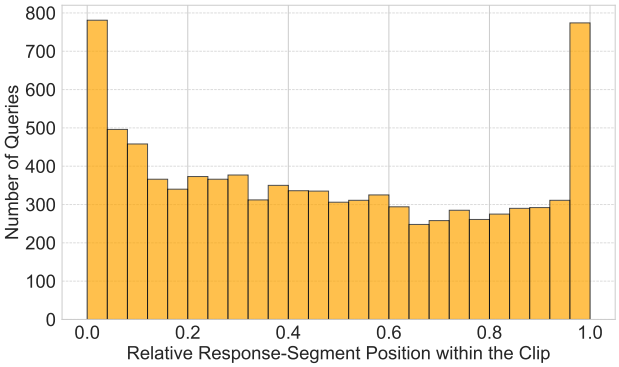

Figure 4. Number of queries by relative Response-Segment position within their clip.

3. Dataset

We will provide here a brief analysis of the Ego4D dataset that we used for our work. First, we will examine the annotations, which convey information related to the textual queries. For the NLQ-type queries related to the Episodic Memory benchmark, we have three files available: nlq_train.json, nlq_val.json, and nlq_test_unannotated.json. These files contain the portions of the dataset designated for the training, validation, and testing phases of the model, respectively. Each file includes a list of queries associated with information about the clip and the reference video. The first two files also provide the ground truth, represented by the start and end times of the response segment.

It should be noted that the following analyses were conducted on both the training and validation datasets, yielding conceptually analogous and consistent results. For brevity, we will refer only to the nlq_train.json file, but similar observations apply to the validation set.

We then extended our preliminary analysis to the ego4d.json file, which contains metadata for the entire dataset, specifically linking each video to various scenarios.

The NLQ annotations are quite heterogeneous: they contain a large variety of queries and videos with different characteristics. The queries can be classified into 13 distinct templates that give the general structure for specific questions (e.g., "Did I leave the window open?" falls under the template "What is the state of object X?"). These templates are further grouped into three categories: *Object*, *People*, and *Place*.

We conducted several analyses, available at https://github.com/Cucchi01/episodic-memory/tree/nlq_fixes_and_fp16_support in the branch **nlq_fixes_and_fp16_support** of the GitHub repository, but we report here only the most relevant ones for brevity.

One of them, in particular, aimed to study the duration of the response segments independently of the template type, to determine the most frequent duration range in the dataset. From Fig. 3, it is evident that most response segments have a relatively short duration, typically falling below 100 frames, which corresponds to approximately 3.5 seconds.

Then, we studied the distribution of the response segments throughout the reference clip to understand which section of the clip had the highest concentration of questions and answers. To achieve this, we assigned a rational number between 0 and 1 to each query, indicating the relative position of the response segment's midpoint within the clip. Subsequently, we computed the number of queries in the dataset that were located close to the same relative position in their respective clips.

By looking at Fig. 4, we can clearly notice how most queries are located towards the beginning and end of the reference clip, with fewer occurrences in the central zone.

Considering the analysis of ego4d.json, instead, we examined only a subset of the Ego4D dataset, focusing on the videos also present in nlq_train.json. The dataset thus obtained includes 136 distinct scenarios, as the whole ego4d.json, and this highlights once again how varied and diverse the NLQ annotations dataset is.

In particular, we found that the most frequent scenarios are Cooking, Cleaning/Laundry, Car mechanic, and Indoor navigation (walking), which are, not by coincidence, situations where it is easier to identify various non-repetitive actions and objects. Due to this favorable nature, more space has been dedicated to them within the dataset.

4. Methodology

In this section, we first present the architectures used to handle the NLQ task, then the process to generate the queries using Gemma, and finally we describe the pre-training and fine-tuning steps.

4.1. Architectures

To tackle the NLQ challenge we used the models *VLS-Base* and *VSLNet* [9], with the latter being an extension of the former and characterized by superior performance thanks to an effective Query-Guided Highlighting (QGH) strategy.
Differently from the original paper, we did not obtain the visual features with a 3D ConvNet, but we used pre-extracted features from Omnivore and EgoVLP. To encode the queries we employed BERT for most of the steps, but we considered also GloVe as an alternative.
In VSLBase, since addressing an NLVL task requires jointly handling textual and visual features and uncovering the interconnections between them, both features are projected into a shared space using two linear layers, unifying the dimensions of the vectors. A single and shared Feature Encoder encodes these features and the Context-Query Attention (CQA) component captures cross-modal interactions. The CQA computes the similarity scores and the attention weights that help to determine how closely related is each visual feature to the relative query feature, refining the representations. Finally, the Conditioned Span Predictor predicts the start and end boundaries of the answer span.
VSLNet builds on top of this structure by adding the QGH. Specifically, it considers the target moment, beginning with a_s and ending with a_e, as the *foreground* and the rest as the *background*. QGH extends the boundaries of the foreground to include its preceding and following video content, where the extension ratio is controlled by a hyperparameter α. By assigning a value of 1 to visual features in the foreground and 0 to those in the background, QGH becomes a binary classification module that estimates the probability that a visual feature is part of the foreground or background.
This last strategy has two significant advantages: firstly, the expanded region offers additional context for identifying the answer span thanks to the continuous nature of video content. Secondly, the highlighted region enables the network to concentrate on subtle differences between video frames, since the search space is at this point significantly smaller compared to the entire video.

4.2. Tests

Before going into detail about the query generation step, we will briefly mention here a few tests we conducted starting from the reference model VSLNet and analyzing the effects and results of not employing the query-guided highlighter, using different visual features, changing the text encoder.
We started by training for 10 epochs both VSLBase and VSLNet twice, first with the pre-extracted visual features from Omnivore, then from EgoVLP, while using the same textual NLQ annotations.
After training, we evaluated the 4 different models on the validation set and on the test set.
Subsequently, we implemented a small variation of VSLNet by replacing the text encoder *BERT* with *GloVe*, whose properties have already been outlined in the previous section.

4.3. Query generation

We created a new and more diverse set of queries related to the entire Ego4D dataset using the available narrations. This data augmentation aims to expose the VSLNet model to a wider range of scenarios during a pre-training phase, before moving on to fine-tuning and using the original, specific training set of NLQ annotations.
The additional queries for pre-training were generated using the *Gemma-2b-it-V2* LLM which was instructed to produce new questions starting from a set of randomly sampled consecutive *narrations* (coming from the narration.json file). These are nothing more than textual annotations covering all Ego4D videos, briefly describing the activity the user is performing at a given moment.

4.3.1 Sampling groups of narrations

To sample the narrations, we started excluding the videos of the Ego4D dataset included in the test and validation sets of the NLQ task to avoid a data leak to the training set.
For simplicity we selected one group of consecutive narrations for each annotation uid, a field with similar properties to clip uid, which is not available in the narrations file, intending to generate new queries covering the selected set of narrations.
Samples of *n* consecutive narrations were taken by randomly selecting the first narration, ensuring that the sample did not present any narration with a "status" field different from "complete" and avoiding those containing the word "unsure" inside the "narration_text" attribute. This operation was performed to facilitate the job of the LLM, which, not receiving incomplete narrations, could provide better quality queries. Additionally, since the pre-trained visual features are based on the clip reference system, sets sharing the same annotation uid but that were not entirely contained within a single clip were also excluded from the sampling.
An important step was the definition of the extremes of the interval of the sample and consequently the ground truth of the future annotations. The input file presents only the

starting time at which each narration is performed, not an interval. From this "limitation" derived our idea to take consecutive narrations and use the timestamps of the first narration in the sample and the one after the last as borders for the response.
Since we already had a good level of variety given by both the free form of the queries generated by the LLM and especially by leveraging a large and diverse set of videos and scenarios (while the annotations of the train set of the NLQ task follow specific templates and cover just a subset of the Ego4D dataset), we aimed to match as close as possible the distribution of the ground truth segment durations of the fine-tuning dataset. In this way, we tried to limit the domain gap between pre-training and fine-tuning datasets. Following this idea, we also avoided creating annotations with very wide response intervals from which the model would have potentially learnt less transferable concepts. We were able to regulate the ground truth duration by modifying the cardinality of the sample of narrations n: the more consecutive narrations the wider the interval.
We therefore conducted a brief analysis of the response segment length in nlq_train.json, finding an average duration of approximately 9.67 seconds. We then observed the average duration covered by the samples of narrations as n varied. Starting from $n = 2$, the average duration was 13.81 seconds, while, for $n = 3$, it was 20.00 seconds, both comparable with nlq_train.json. Additionally, we also considered the configuration starting from $n = 5$ narrations, which had an average duration of the response interval considerably higher than the original NLQ set (about 31.64 seconds), to be able to analyze the impact of different lengths of ground truth on the results. To have a fair comparison among the three configurations and at the same time a number of queries similar to the one of the original annotations set, we decided to produce in all three cases 8500 queries. This value was obtained empirically by checking that all three configurations were always able to produce a quantity of queries slightly greater than this value, which is close enough to the number of queries of nlq_train.json. To achieve this, we instructed the LLM to provide, respectively, $m = 2$, 2 and 3 queries.

4.3.2 LLM prompt

To obtain a good level of quality for the queries generated by the LLM, we manually tested different prompts and output lengths. The three main prompts that we tested are:

A) "You are trying to help humans to augment their memory. To achieve this goal some annotators have watched several videos about people performing different actions in various scenarios, writing down, step by step, short narrations describing in a few words what is happening in the video and the time at which these actions take place. You are requested to generate 2 simple queries that a person could answer just by looking at the video segments corresponding to these narrations:" + formatted_narrations[i] + " To do so follow these rules: - do not include the answer in the question - avoid introductions, report just the text of the queries one per line"

B) "Generate two short simple queries that a person could answer looking at the video corresponding to these narrations: "+ formatted_narrations[i]+ "\nIn your answer report just the queries one per line"

C) "Generate two questions mixing the information of these sentences: \""+ formatted_narrations[i]+ "\"\nIn your answer write only two lines with a question each"

Running several tests we were able to notice that the simpler the prompt the better-quality were the queries obtained. Here is an example:

Narration: "C fixes the camera. C walks in the supermarket. C picks the shopping basket"

• Result prompt A: ['What is the person doing at 2 minutes and 40 seconds into the video?']
• Result prompt B: ['What is the action that C does in the video?', 'What does C do in the supermarket?']
• Result prompt C: ['Where did C fix the camera?', 'What was C doing while walking into the supermarket?']

This behavior is probably related to the fact that we used a light version of Gemma, which, not being as powerful as the latest LLMs, struggled with long and detailed prompts. For this reason, in the end, we employed the prompt C.
To decide the output length in tokens of the model, we operated a trade-off between the speed of generation and the number of queries generated. Different lengths of output did not have a significant influence on the quality of the queries, but in our tests we were able to appreciate how the longer the output the longer was the generation process. On the other hand, however, when we adopted a too short output length, the LLM was often not able to terminate the queries, which had to be filtered out because incomplete, reducing the final number of producible annotations. For these reasons, we sought the shortest output with a negligible number of incomplete queries which resulted in being 20 tokens for each query generated.

4.4. Pre-training and fine-tuning

We will now discuss how we managed the pre-training and fine-tuning phases. In both circumstances, we fine-tuned on the original NLQ annotations dataset for 10 epochs

to have a direct comparison with the results of VSLNet without pre-training. During pre-training VSLNet was trained on the 8500 newly generated queries, while for the fine-tuning we trained the model on the nlq_train.json set, testing two different versions of fine-tuning. Specifically, we experimented firstly by updating the parameters of all layers and secondly by freezing some layers. In the latter case, the achievable results will indicate the generality of the concepts learnt during pre-training.
In the second version, we decided to freeze until the CQA block, leaving the remaining modules trainable because the CQ-Attention module captures cross-modal interactions, which are transferable concepts that can be effectively learned during pre-training. In contrast, the QGH module and the Conditioned Predictor are more specifically tailored to the NLQ task and, thus, we kept them trainable during fine-tuning to better adapt to the specific job.

In Sec. 5, we will present the results obtained for all the different settings: varying *n*, *m* and the type of fine-tuning applied.

5. Experiments

In this section we will present all the results obtained from the tests we conducted and described in the previous sections.

5.1. Evaluation metrics

As evaluation metric, we adopted "recall@k, IoU= t", following the video-language grounding literature. This metric denotes the percentage of times where at least one of the top-k predicted moments has an intersection-over-union (IoU) with the ground truth greater or equal to t. In our case, we report the results for k = 1,5 and t = 0.3, 0.5.

5.2. Baselines

Our goal is to evaluate the effect of our extension combined with existing methods present in the literature. For this reason, we first evaluated the baselines for the task, to be able to have a measure of comparison for our results.
In Tab. 1, we report the results we obtained on the validation and test set comparing VSLNet and VSLBase which are given as input the visual features generated by Slowfast [2] (official NLQ baseline provided by Ego4D, see details in [4]), Omnivore and EgoVLP. As expected and explained in [9], VSLNet, thanks to the query-guided highlighter module, performs better than VSLBase across all metrics. The performance gap is more evident for the visual features obtained with EgoVLP.
Regarding the visual feature extractors, VSLNet reaches better results receiving the video features from EgoVLP than from Omnivore or Slowfast. This is due to the fact that, while Omnivore is a general-purpose model designed to handle a large variety of visual tasks and Slowfast is primarily designed for general video action recognition, EgoVLP, instead, is specifically designed for egocentric video-language pre-training.

Vis. Feat	Model	IoU = 0.3 (%)		IoU = 0.5 (%)	
		r@1	r@5	r@1	r@5
Validation					
SlowFast	VSLNet	5.45	10.74	3.12	6.63
Omnivore	VSLNet	6.22	13.19	3.54	8.18
	VSLBase	5.58	12.13	2.76	7.77
EgoVLP	VSLNet	**7.90**	**15.75**	**5.06**	**10.33**
	VSLBase	5.63	12.70	3.36	8.26
Test					
SlowFast	VSLNet	5.47	11.21	2.80	6.57
Omnivore	VSLNet	6.56	11.26	3.67	6.89
	VSLBase	6.54	10.96	3.84	6.61
EgoVLP	VSLNet	**8.66**	**13.68**	**5.26**	**9.09**
	VSLBase	5.64	10.66	3.47	6.81

Table 1. Results obtained by VSLNet and VSLBase models, compared to the baseline using Slowfast provided by Ego4D, trained on different sets of visual features and with BERT.

5.3. Variation of Text Encoder

Tab. 2 contains the results obtained with two different text encoders: BERT and GloVe. As clearly shown in the table BERT significantly outperforms GloVe across all metrics, leveraging contextual embeddings, bidirectional representations, and a sophisticated transformer architecture exploiting self-attention mechanisms. GloVe, on the other hand, uses static embeddings and is more focused on word co-occurrence patterns than on the actual semantic meaning in the context. However, thanks to its simpler approach, GloVe is faster than BERT: measuring the elapsed time for the training with the two configurations of VSLNet we obtained 10 minutes and 23 seconds using BERT and 9 minutes and 32 seconds employing GloVe. Nonetheless, considering the total duration of the process, the advantage in time is not that significant to lead us to choose GloVe over the better-performing BERT.
For the final step of the analysis, we adopted the best performing configuration: VLSNet with EgoVLP features and BERT text encoder.

5.4. Automatic queries generation with LLM

In this final section we will present the results obtained by pre-training the VSLNet on a set of annotations containing the queries automatically generated with Gemma from the dense narrations provided by Ego4D. They are summarized in Tab. 4.
As described in the methodological section, once we had pre-trained the VSLNet using the new set of annotations, we conducted two experiments: in one case we fine-tuned all layers of our model, in the other one we froze the model's

Configuration	IoU = 0.3 (%)		IoU = 0.5 (%)	
	r@1	r@5	r@1	r@5
Validation				
VSLNet & BERT	**6.22**	**13.19**	**3.54**	**8.18**
VSLNet & Glove	3.05	8.36	1.21	4.34
Test				
VSLNet & BERT	**6.56**	**11.26**	**3.67**	**6.89**
VSLNet & Glove	3.54	7.21	1.72	4.02

Table 2. Results obtained by VSLNet model, trained on the Omnivore visual features, with BERT and Glove as text encoders, on the validation set and the test set.

parameters until the context query attention layer.

Even if the number of epochs for which we pre-trained our model was limited (only 10, due to limited resources), the model is learning from the generated queries, as shown by the recall rates after the pre-training phase in Tab. 3. Our approach, when we let then all layers free and fine-tune the model for 10 epochs, reaches better results across all metrics, outperforming the baselines.

These results confirm our hypotheses: by exposing the model to a wider range of linguistic structures and visual contexts it is possible to enhance its performance.

The results also highlight the impact of different fine-tuning strategies: fine-tuning all layers after pre-training shows better performance than freezing the initial layers and only updating the last two. This indicates that the model benefits more from a holistic adjustment to the new task-specific data. When only the last few layers are fine-tuned, a significant part of the model remains fixed with its general pre-training knowledge. In this case, not being able to adjust the parameters of the first layers to better align with the new task leads to suboptimal performance, probably because the gap between pre-training and fine-tuning datasets is significant, since in the first case we are dealing with free-form automatically generated queries.

Another possible explanation is that our limited number of epochs and queries, due to the few resources, are not sufficient to fully capture the knowledge and features that LLM-generated queries and videos can provide, on top of which fine-tuning only the last layers could be more beneficial.

For what concerns the different experiments conducted varying the number of narrations from which we generated the queries, as shown in Tab. 4, no clear difference is evident in the results. This suggests that pre-training the model using ground truths of different lengths does not have a significant impact at fine-tuning time (for the mechanism of extraction of narrations and the way in which we compute the relative ground truth, on average the amplitude of the interval to identify increases when the number of narrations increases).

The most significant limitation of this approach is probably

Pre-training	IoU = 0.3 (%)		IoU = 0.5 (%)	
	r@1	r@5	r@1	r@5
Before	1.21	2.89	0.44	1.21
After	2.84	6.22	1.45	3.69

Table 3. Results obtained on the validation set by the VSLNet model with BERT and EgoVLP features, before and after pre-training on different sets of two new queries generated by *Gemma* from samples of three consecutive narrations.

Configuration	IoU = 0.3 (%)		IoU = 0.5 (%)	
	r@1	r@5	r@1	r@5
Best baseline	7.90	15.75	5.06	10.33
Updating all layers				
n=2, m=2	8.80	**17.27**	**5.50**	11.20
n=3, m=2	8.65	**17.27**	5.06	11.38
n=5, m=3	**9.09**	16.88	5.32	**11.49**
Freezing some layers				
n=2, m=2	**4.03**	**9.65**	2.48	5.83
n=3, m=2	3.87	9.53	2.01	**5.96**
n=5, m=3	3.98	9.16	**2.56**	5.65

Table 4. Results obtained on the validation set by the VSLNet model with BERT and EgoVLP features, pre-trained on different sets of m new queries generated by *Gemma* from samples of n consecutive narrations, and fine-tuned on the original NLQ annotations. In the second part, the results are obtained freezing the first layers and updating only the last 2 layers.

the inability to assess the quality of the generated queries automatically. While the set of annotations provided by Ego4D is generated manually, following specific indications and rules, generating queries with an LLM can provide unpredictable results, especially if, like in our case, a light version is used. However, even with a "smarter" LLM, capable of always generating sensible and pertinent queries, some problems still arise: for instance there is no way to instruct the LLM, which just receives few narrations, to avoid generating queries whose answer could be contained also in another segment of the video, situation that is specifically avoided in the case of the manual annotations.

6. Conclusions

To summarize, we focused on the Episodic Memory benchmark and used state-of-the-art models and new methods for data augmentation and pre-training.

First, we compared the performance of VSLNet and VSLBase models using two sets of visual features: Omnivore and EgoVLP. We found that VSLNet, especially with EgoVLP features, consistently outperformed VSLBase across all metrics.

Next, we tested different text encoders on VSLNet: BERT performed considerably better than GloVe, showing the benefits of contextual embeddings for understanding

natural language queries.

To further improve the results, we created an automatic query generation process using the *Gemma-2b-it-V2* large language model. Pre-training VSLNet on these generated queries led to noticeable improvements, especially when we fine-tuned all layers of the model. This confirmed that exposing the model to a wider range of language structures enhances its performance.

Finally, we observed that different strategies for freezing model layers during fine-tuning have different impacts on performance. Fine-tuning all layers after pre-training gave better results than freezing the initial layers, indicating that leaving the model free to adapt to new task-specific data is important for optimal performance in our case.

Overall, our findings suggest that a combination of pre-extracted visual features, contextual text embeddings, and extensive pre-training on automatically generated queries can significantly improve the performance of our models on NLVL tasks in egocentric video datasets.

Future work may explore further optimization of the query generation process and fine-tuning strategies to continue improving first-person video understanding without the restrictions on the number of generated queries and pre-training epochs imposed by our limited resources.

Code available at: https://github.com/Cucchi01/episodic-memory/tree/nlq_fixes_and_fp16_support.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 2

[2] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition, 2019. 6

[3] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities, 2022. 2

[4] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. 2022. 1, 6

[5] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, Hongfa Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining, 2022. 2

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01 2014. 2

[7] Santhosh Kumar Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Naq: Leveraging narrations as queries to supervise episodic memory, 2023. 2

[8] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. 2

[9] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization, 2020. 2, 4, 6

[10] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language, 2020. 2