Natural Language Queries for Egocentric Vision

Linda Öqvist
s347769

Mohamad Parichehrehteroujeni
s328173

Luca Desderi
s347876

Abstract

This project explores how to understand egocentric videos using natural language queries. The main goal was to train models to accurately localize a video segment that answers a given question. To do this, the Ego4D dataset and its NLQ (Natural Language Query) benchmark task was used. A comparative study of two prominent models, VSLBase and VSLNet, was conducted using visual features from Omnivore and EgoVLP respectively. For the input queries, two different methods for encoding text were compared and BERT was identified as the most effective one. As an extension to the project, new training data was generated using a large language model, Gemma-2b, based on video narrations that are not part of the original NLQ benchmark. It was found that using these automatically generated queries for pretraining VSLNet significantly boosts the model's performance upon fine-tuning on the official Ego4D NLQ data. Overall, the results show that combining VSLNet with EgoVLP features and BERT, along with pretraining on automatically generated queries, leads to more accurate understanding of egocentric videos through natural language. The code is available at `https://github.com/mhmdparichehreh/egocentric-vision.git`

1. Introduction

Egocentric vision, also called first-person vision, is a field belonging to computer vision that aims to analyze images and videos captured by a wearable camera. Current computer vision systems perform well on image and video datasets that contain short isolated moments from a third-person view. However, despite recent advances in general video understanding, the application of accurately understanding egocentric footage using natural language queries remains relatively underexplored. In for example robotics and augmented reality, the system has to analyze long, fluid videos from a first-person or "egocentric" point of view. Another factor that differentiates egocentric vision from usual computer vision tasks is that the footage from a wearable egocentric camera, that always is turned on, lacks the curation of most other footage. To understand a first-person perspective, one needs a 3D understanding of the physical surroundings of the camera-wearer, and it also requires interpretation of objects and actions in a human context [7]. These factors make egocentric vision an important field of research.

This project focuses on the task of NLVL (Natural Language Video Localization) in egocentric videos using the NLQ benchmark in the Ego4D dataset [7]. The first part of the project consisted of training a model, VSLNet (Video Span Localizing Network) [11], to localize the segment of an egocentric video that answers a given natural language query. The second part of the project was to automatically generate new queries on videos in Ego4D that are not present in their NLQ data. This was done using the narrations on the Ego4D videos and a Large Language Model (LLM). These new queries were then used to pretrain the model.

Figure 1. Some frames of egocentric videos taken from Ego4D [7].

2. Related work

The task of NLVL is to locate the span in an untrimmed video that semantically corresponds to a given text query. One proposed solution to this task is to use a span-based QA (Question Answering) approach, which is based on treating the input video as a text passage. Two different architectures using this approach, VSLBase and VSLNet, is introduced by [11]. VSLBase is a standard span-based QA framework that they use as a baseline. However, there is a problem with this span-based QA approach, namely that there are two main differences between traditional text span-based QA and NLVL.

First, video is continuous, and events that are related often happen right next to each other in time. In contrast, natural language follows a syntactic structure and related words may be far from each other. For example, while adjacent video frames are usually very similar, neighboring words in a sentence can have very different meanings. Second, people are generally less sensitive to small changes in video frames than they are to changes in a text. A slight shift in video frames usually does not affect how we understand the video, but changing even one word in a sentence can significantly alter its meaning [11].

VSLBase does not address these two major differences between video and natural language. Therefore they also propose an improved version of the network called VSLNet, on top of the standard span-based QA framework. The proposed VSLNet tackles the differences between NLVL and span-based QA through a simple and yet effective QGH (Query-Guided Highlighting) strategy. The QGH guides VSLNet to search for a matching video span within a highlighted region. The authors present the results from experiments with VSLNet on three benchmark datasets and showed that the model outperforms the state-of-the-art methods. For example, they achieved 63.16% accuracy (r@1, IoU=0.3) on the ActivityNet Caption dataset, a dataset containing a wide range of complex human activities in 200 different classes [11]. However, none of the datasets they tested VSLNet on contains egocentric videos.

Another approach to action classification in videos is to use a network with one slow path and one fast path, a so called SlowFast Network, introduced by [5]. The slow path extracts detailed spatial and semantic information from the video frames but the frames are sampled at a very slow temporal rate. The fast path has a higher temporal frame rate and is used to extract motion information. However, the fast path has a lot fewer channels than the slow path making it lightweight (i.e. not computationally expensive). The information of the two pathways are then fused by lateral connections. Results of their experiments with the SlowFast network show that it achieves strong performance for both action classification and action detection in video. For action classification they achieved 79.8% top-1 accuracy on the Kinetics-400 dataset, a dataset containing a broad range of human activities divided into 400 action classes.

Training models end-to-end on video datasets is often very computationally expensive and time consuming. A common strategy to save time and improve the performance is therefore to pretrain it and then finetune it for the specific downstream task we want it to solve. In this project, two pretrained models will be tested, EgoVLP [8] and Omnivore [6].

Video-Language Pretraining (VLP) has prevailed as a way to improve video-text tasks, such as video question answering, by learning a strong and transferable video-language representation. To bridge the domain gap between the existing video-text pretraining datasets and videos from an egocentric view, an egocentric video-language pretraining dataset has been developed by [8]. It is called EgoClip and contains a total of 38M clean first-person clip-text pairs selected from the Ego4D dataset. By conducting experiments they demonstrated that the performance on three Ego4D benchmark challenges, including NLQ, could be boosted by using their EgoVLP (Egocentric VLP).

Most computer vision architectures are specialized for recognition of either images, videos or 3D data. Instead the Omnivore model, proposed by [6], is a single model which has high performance at classifying images, videos, and single-view 3D data using exactly the same model parameters. The Omnivore model has the flexibility of transformer-based networks and can therefore be trained jointly on classification tasks from different modalities. When tested on the task of action classification the model obtained 83.4% top-1 accuracy on the Kinetics-400 dataset and 47.4% top-1 accuracy on the EPIC-Kitchens-100 dataset [3]. EPIC-Kitchens-100 consists of egocentric videos of kitchen activities. Omnivore presents an advance over traditional modality-specific models.

In Table 1 the baselines of VSLNet's performance on the Ego4D NLQ benchmark task from [7] and [8] are presented. These will be used for comparison with the results obtained from this project. The results from the Ego4D paper [7] were produced using pre-extracted features for both the video clip, using SlowFast network [5], and natural language query, using a frozen pretrained BERT model [4]. They also used a learning rate of 0.0001, with linear decay, and trained the model for 200 epochs. The results from the EgoVLP paper [8] were produced by replacing the SlowFast-BERT features used in [7] with their video and language representations. They adopt Frozen [1], pretrained on their EgoClip data, together with their video-text training objective EgoNCE (NCE: Noise Constrastive Estimation), via positive and negative sampling for egocentric-friendly pretraining. They used a learning rate of 3×10^{-5} and pretrained the model for 10 epochs.

3. Methodology

In this section the pipeline of the project will be described, as well as the network and the dataset that was used. The evaluation metrics used to evaluate the results of the project will also be presented.

Baseline	IoU@0.3 (%)		IoU@0.5 (%)	
	r@1	r@5	r@1	r@5
VSLNet, Ego4D paper	5.45	10.74	3.12	6.63
VSLNet, EgoVLP paper	10.84	18.84	6.81	13.45

Table 1. The baseline performance of VSLNet on the Ego4D NLQ benchmark task from the Ego4D paper [7] (top row) and from the EgoVLP paper [8] (bottom row).

3.1. Pipeline

This project consisted of three main parts, of which the first part was to get familiar with the concept of egocentric vision and NLQ by reading literature on the topic. This first step also consisted of getting familiar with the Ego4D dataset [11] that was used in the project.

The second step of the project was to train a network called VSLNet and its less complex version VSLBase on the NLQ benchmark of the Ego4D dataset [11]. This was done in Google Colab with the help of this repository: `https://github.com/EGO4D/episodic-memory`. In order to use VSLBase instead of VSLNet some parts of the code had to be removed. Both versions of the network were trained with two different sets of pre-extracted video features to make a comparison, Omnivore [8] and EgoVLP [6]. A comparison was also made between two different text encoders, first the network was trained with the text encoder BERT (Bidirectional Encoder Representations from Transformers.) [4] and then with 300d GloVe (Global Vectors) word embeddings [9] instead. The results from all these different variations of the network were compared with each other and with the baseline provided in the Ego4D paper [7] and in the EgoVLP paper [8].

The third and last step of the project was the extension. The extension that was chosen was the first one proposed for the project, to generate new queries automatically using the narrations of the videos in the dataset and a LLM (Large Language Model). These new queries were then used to pretrain VSLNet to see if its performance on the original NLQ data could be improved by this pretraining.

The extension was carried out by first extracting the narrations from all videos, excluding those present in the NLQ benchmark data. Then three consecutive narrations were randomly sampled from each of those videos. These narrations were then fed to a LLM called Gemma-2b [10], a lightweight, state-of-the art open LLM that demonstrate strong performance across academic benchmarks for language understanding and reasoning. The prompt to Gemma asked it to generate one query for each three consecutive narrations, that could be answered by looking at the video segment corresponding to those narrations. This resulted in one query per video and after filtering out some videos that were not present among the EgoVLP features it ended up being 6885 new queries. With these queries a new JSON file was created with almost the same format as the original NLQ data. The ground truth, the start and end time for the video segment answering the query, was created by first taking the timestamp of the first narration and the timestamp of the third and last consecutive narration. Then 2 seconds were subtracted from the first timestamp and 2 seconds were added to the third timestamp. The resulting times were then set as start time and end time of the answer video segment. The 2 extra seconds at the start and at the end were derived from the fact that there on average are 13.2 sentences (each narration in the dataset is one sentence) per minute of video in the Ego4D dataset [7]. This means that each narration describes approx. 4.5 seconds of video on average. Since the timestamp of a narration mark the middle of the narrated action this results in each narration describing a video segment that on average spans from roughly 2 seconds before its timestamp to around 2 seconds after it.

In order to then pretrain VSLNet on this new JSON file with new augmented NLQ data some changes had to be done in the code for VSLNet. The scripts `prepare_ego4d_dataset.py`, `data_gen.py` and `main.py` had to be altered to only use training data (no validation or test data) and to save the model parameters directly after the training phase. Then `main.py` had to be altered again to load the pretrained weights before starting the training phase.

The hyperparameters that were used in all model training during the project, unless otherwise stated, are presented in Table 2. A linear decay schedule with warm-up was used on the learning rate. This schedule starts with a warm-up phase where the learning rate increases linearly from zero to the set value, over a chosen number of warm-up steps. Then there is a decay phase where the learning rate linearly decreases from the set value to zero over the remaining steps of the training.

Hyperparameter	
Number of epochs	10
Initial learning rate	0.0025
Batch size	32

Table 2. The hyperparameters used in training of the model.

3.2. Dataset and Task

This project is focused on the Ego4D dataset, which is an egocentric video dataset and benchmark suite. [7]. It contains over 3000 hours of unscripted video of daily life activity with a wide range of different scenarios (household, outdoor, workplace, leisure, etc.). The videos were captured by 931 individual camera wearers and from 74 locations spread over 9 different countries and 5 continents. The dataset was collected with the aim of diversity of the videos. The camera wearers have a wide variety of occupations and belong to many different age groups. Moreover, among the camera wearers there were almost the same number of women as there were men. The entire dataset is narrated, which means that there are sentences explaining every action that occurs in the videos.

Ego4D also presents five benchmark tasks and the one this project is focused on is the NLQ part of the Episodic Memory task. The scale of the NLQ dataset is presented in Table 3. The Natural Language Queries are expressed in text, for example “Where did I put the watering can?”, and the ground truth is the temporal window in the video where the answer is visible or can be deduced. The queries can be related to objects, places, people, and performed activities. They are defined based on a set of 13 templates, shown in Table 4. In the NLQ training data around 8900 queries belong to the ”Objects” category, around 1700 queries belong to the ”Place” category and around 6000 queries belong to the ”People” category.

Split	Train	Val	Test
# video hours	136	45	46
# clips	1.0k	0.3k	0.3k
# queries	11.3k	3.9k	4.0k

Table 3. The NLQ dataset statistics across the training/validation/test splits [7].

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?

Table 4. The different types of query templates present in the NLQ data of Ego4D [7].

NLQ is a challenging multimodal task, as it requires both textual and visual understanding and reasoning. Another difficulty of the task is that videos may be of arbitrary length and it therefore needs a model that can reason over long temporal ranges.

The clips in the NLQ train split are around 522 seconds long on average while the average length of the query answer segments are around 9 seconds. In Fig. 2 two histograms over relative query sizes in the NLQ training data are shown. The relative query size is the length of the video segment that answers the query divided by the length of the clip in which the model has to locate the answer segment. In Fig. 2a the relative query sizes smaller than 0.2 are shown and in Fig. 2b all relative query sizes larger than 0.2 are shown. It can be seen that the majority of the queries has a relative query size smaller than 0.2, which means that the answer segment is usually a lot shorter than the entire clip, making the NLQ task more difficult.

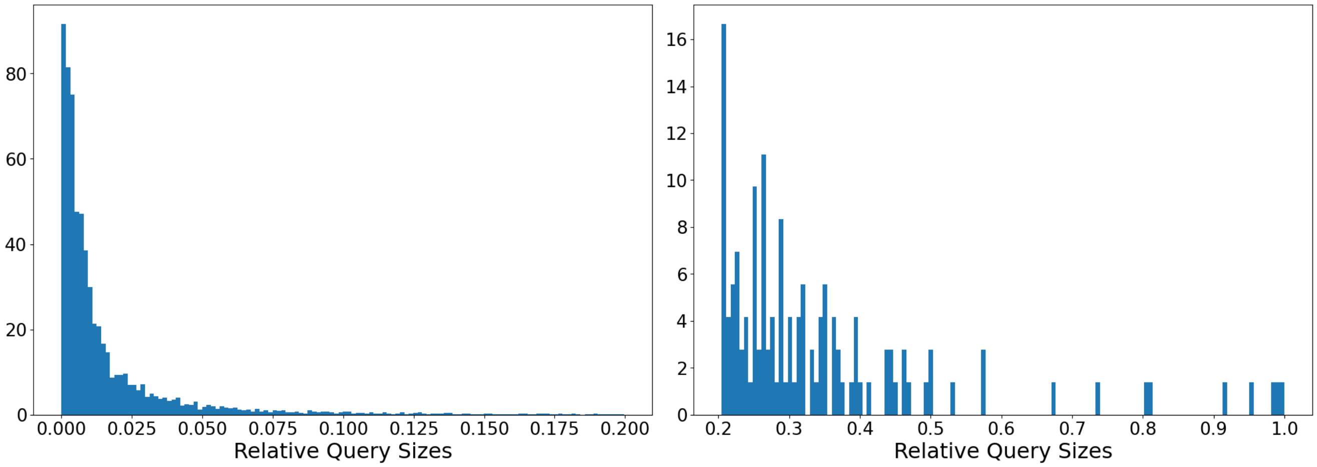
3.3. Network details

The networks used in this project are VSLBase and VSLNet, proposed by [11]. The structure of these two networks can be seen in Fig. 3. VSLBase is a standard span-based QA framework that treats the input video as a text passage and the target moment is regarded as the answer span. VSLNet differs from VSLBase by the addition of a QGH layer, which guides VSLNet to search for a matching video span within a highlighted region.

VSLBase starts with two branches that extract visual and textual features from the input video and the query text respectively (blue and red blocks Fig. 3). VSLBase then encodes the natural language query and video features using a common, shared encoder block (yellow blocks in Fig. 3). This encoder block consists of four convolution layers, followed by a multi-head attention layer. A feed-forward layer is used to produce the output and layer normalization and residual connection are applied to each layer. After the feature encoding, a CQA (Context-Query Attention) is used to capture the similarity between the visual and textual features (violet block in Fig. 3). Then follows a conditioned span predictor that uses two unidirectional LSTMs (Long Short Term Memory) and two feed-forward layers to regress the temporal boundaries of the answer (green block inFig. 3). The two LSTMs are stacked so that the LSTM of the end boundary can be conditioned on that of the start boundary. Then the hidden states of the two LSTMs are fed into the corresponding feed-forward layers to compute the start and end scores. In this way VSLBase is trained to predict the start and end boundaries of the answer span. The training objective is defined as:

$$\mathcal{L}_{\text{span}} = \frac{1}{2}[f_{CE}(P_s, Y_s) + f_{CE}(P_e, Y_e)] \tag{1}$$

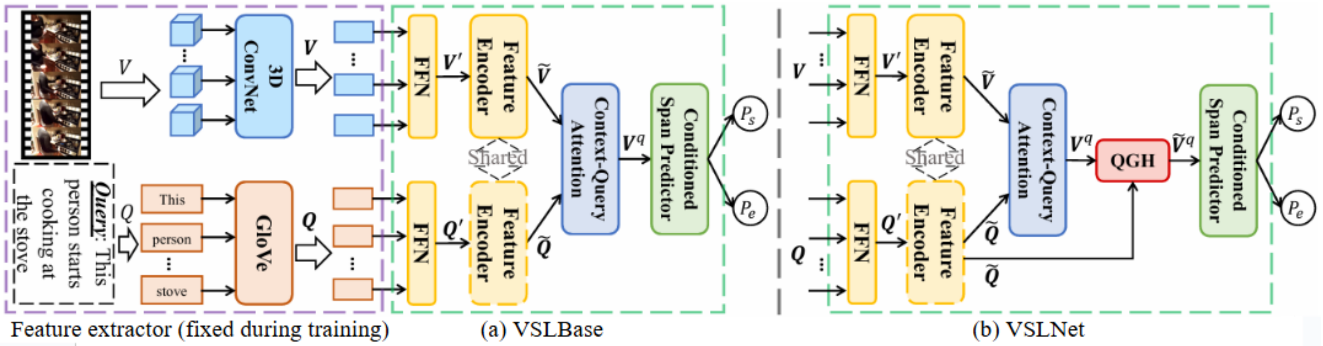
where f_{CE} represents cross-entropy loss function, Y_s and Y_e are the labels for the start and end boundaries and P_s and P_e are the probability distributions of the start and



(a) Smaller than 0.2.

(b) Larger than 0.2.

Figure 2. Histograms over relative query sizes in the NLQ data of the Ego4D dataset [7].



Feature extractor (fixed during training)

(a) VSLBase

(b) VSLNet

Figure 3. The structure of VSLBase and VSLNet. (a) depicts the structure of VSLBase and (b) shows the structure of VSLNet. Figure from [11].

end boundaries.

VSLNet is an improved version of VSLBase, where a QGH layer has been added to address the differences between text span-based QA and NLVL tasks, described in Section 2. The QGH strategy works by considering the target moment as the foreground, and the rest as background. QGH then extends the boundaries of the foreground to cover video parts before and after the target moment. The extension ratio is controlled by a hyperparameter α. The idea of this extended boundary is to potentially cover additional contexts and help the network to focus on subtle differences between video frames. By assigning 1 to foreground and 0 to background, a sequence of 0-1 is obtained, denoted by Y_h. The QGH is a binary classification module to predict the confidence of a visual feature belonging to foreground or background. First the word features $\tilde{\mathbf{Q}}$ are encoded into sentence representation (denoted by h_Q), with a self-attention mechanism. Then h_Q is concatenated with each feature in $\mathbf{V}^q$, which is the output of the CQA. The result of the concatenation is denoted $\bar{\mathbf{V}}^q$ and a highlighting score is computed as:

$$\mathcal{S}_{\text{h}} = \sigma(\texttt{Conv1D}(\bar{\mathbf{V}}^q)) \tag{2}$$

and is then used to calculate the highlighted features as $\tilde{\mathbf{V}}^q = \mathcal{S}_{\text{h}} \cdot \bar{\mathbf{V}}^q$. The loss function of query-guided highlighting is formulated as:

$$\mathcal{L}_{\text{QGH}} = f_{CE}(S_h, Y_h) \tag{3}$$

VSLNet is then trained by minimizing the following loss:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}} \tag{4}$$

When training the network the optimizer used was AdamW. AdamW is an extension of the Adam (Adaptive Moment Estimation) optimizer, that incorporates weight decay directly into the optimization process.

3.4. Evaluation metrics

Following the evaluation metrics used in the Ego4D paper [7], performance is evaluated in terms of top-k recall (r@k) at a certain temporal Intersection over Union (tIoU) threshold. The tIoU compares a predicted time interval with the ground truth time interval and therefore it measures how much the interval predicted by the network overlaps with the correct interval. When using top-k recall the best k predictions of the model are analyzed, and the recall is 1 if at least one of those predictions matches the ground truth with tIoU over the set threshold.

In this project k = 1, 5 were used and the IoU thresholds used were 0.3 and 0.5. This metric computes the percentage of times that at least one of the top predicted candidates has an IoU equal to or higher than the threshold. The reason for the threshold values being rather low is that the average length of the window (10s) is much smaller than that of the video clip (500s), about 2% of the clip length.

4. Experiments and results

This section presents the results from the different experiments conducted during the project. The first part contains the results from training different variations of the model and in the second part the results from the extension are presented, where new queries were automatically generated and then used to pretrain the network.

4.1. Variations of the network

The results from training different variations of the model are shown in Table 5. The ”Properties” column shows which version of the network, which video features, and which text encoder that was used. The highest results were obtained with IoU threshold of 0.3, pre-extracted features from EgoVLP [8] and a pretrained BERT model [4] as text encoder. These results are highlighted in the table.

Properties	IoU@0.3 (%)		IoU@0.5 (%)	
	r@1	r@5	r@1	r@5
VSLNet, EgoVLP, BERT	**6.81**	**14.30**	4.10	9.53
VSLNet, Omnivore, BERT	6.48	13.53	3.51	8.29
VSLNet, EgoVLP, GloVe	6.56	13.11	3.87	8.67
VSLBase, EgoVLP, BERT	6.09	12.67	3.95	8.36
VSLBase, Omnivore, BERT	5.45	12.03	3.20	7.77

Table 5. The performance of VSLNet and VSLBase on Ego4D NLQ benchmark, with different video features and different text encoders. The highest results are written in bold.

4.2. Automatic generation of new queries for pretraining VSLNet

The new queries that were generated by the LLM Gemma-2b were of varying quality, but overall they turned out moderately good. The queries did not turn out as diverse as in the original NLQ data, nor as fitting for the corresponding video segment. Two examples from the generated queries are presented in Table 6. The first query in the table is more precise and more relevant to the narrations than the second one, which shows the varying quality of the generated queries.

In Table 7 the results of VSLNet, with and without pretraining on the automatically generated query data, is presented. Here VSLNet was used with EgoVLP features and BERT. In order to better showcase the effect of the pretraining, the results are presented both after one epoch and after ten epochs. The performance on the original NLQ data is higher already after epoch 1 and also after epoch 10, which shows that pretraining on the new query data had a positive effect.

The VSLNet model, with EgoVLP features and BERT, was also trained for 15 epochs with and without pretraining, which gave better results after epoch 10, see Table 8, than when epoch 10 was the last epoch, see Table 7.

5. Discussion and conclusion

The aim of this project was to experiment with VSLNet on the task of NLQ in egocentric videos, specifically on the Ego4D dataset. Different variations of the network were trained to show how these changes in the model affect performance. The possibility of extending the NLQ data with automatically generated queries was also investigated, as well as the effect of pretraining VSLNet on such queries.

5.1. Variations of the network

VSLNet showed slightly better performance than VSLBase (see Table 5) which was expected, since VSLNet has an extra layer, the QGH, that guides the network to search for a matching video span within a highlighted region. VSLNet outperforming VSLBase is in line with the results presented in the paper that introduces these networks [11].

Using BERT as text encoder resulted in better performance of VSLNet than when GloVe word representations was used (see Table 5). This is expected as BERT is a much more advanced model than GloVe for encoding text. Firstly, GloVe contains non-contextual embeddings while BERT contains contextual ones, which means that BERT is better at capturing the context of the NLQs [2]. Secondly, BERT is pretrained on a large collection of texts and therefore outperforms GloVe at encoding the semantics of the queries [4, 9]. These are some of the factors that make

Narrations	Generated query
#C C puts on a tap. #C C picks up a spoon from a sink. #C C rinses the spoon.	What is the person doing with the spoon after picking it up from the sink?
#C C holds a phone. #C C looks at the phone. #C C takes away the phone.	What is the person holding in their hands?

Table 6. Two examples of queries generated by Gemma-2b, given three consecutive narrations.

Network (# epochs)	IoU@0.3 (%)		IoU@0.5 (%)	
	r@1	r@5	r@1	r@5
VSLNet (1)	1.29	5.11	0.80	3.23
VSLNet, pretrained (1)	5.99	12.62	3.90	7.82
VSLNet (10)	6.81	14.30	4.10	9.53
VSLNet, pretrained (10)	**9.42**	**17.22**	5.34	11.33

Table 7. The validation results, after epoch 1 and after epoch 10, of VSLNet on the NLQ data with and without pretraining on the new queries. Model trained for a total of 10 epochs.

Network (# epochs)	IoU@0.3 (%)		IoU@0.5 (%)	
	r@1	r@5	r@1	r@5
VSLNet (10)	8.52	16.57	4.93	10.82
VSLNet, pretrained (10)	9.73	17.94	5.94	11.85

Table 8. The validation results after epoch 10 of VSLNet on the NLQ data with and without pretraining on the new queries. Model trained for a total of 15 epochs.

BERT better at understanding queries, matching text with video features and generalizing to the complex egocentric data.

Both VSLBase and VSLNet performed better with EgoVLP video features than with Omnivore video features (see Table 5). This is not surprising since EgoVLP is designed specifically for egocentric video-language tasks while Omnivore is a vision model for general purposes. Moreover, EgoVLP uses video-language contrastive learning, video-text matching, and temporal localization, while Omnivore primarily focuses on classification [6, 8]. This makes EgoVLP more suitable for the NLVL task than Omnivore. However, both gave better results than the baseline from the Ego4D paper (see Table 1), where VSLNet was trained with SlowFast video features instead. That Omnivore features outperform SlowFast features is expected since omnivore achieved a higher performance on the Kinetics-400 dataset than SlowFast did, see Sec. 2. The performance of VSLNet achieved in this project is good compared to this baseline. However, when comparing the results of this project with those achieved with VSLNet in the EgoVLP paper (see Table 1), ours seem a bit low. There are several possible explanations for this, where one is that they used a different method for encoding visual and text features. They used Frozen, pretrained on EgoVLP features, together with their pretraining objective EgoNCE for both the video and text branch, while we used their EgoVLP features for the video branch only and then BERT for the text branch. That is probably the principal reason to their results being higher than ours, but it could also have to with differences in the hyperparameters used.

5.2. Automatic generation of new queries for pretraining VSLNet

This extension of the project was successful, since the performance of VSLNet increased significantly after pretraining on the new NLQ data with the automatically generated queries (see Table 7). The model setup used in this part was VSLNet with EgoVLP video features and BERT as text encoder, since that setup had achieved the highest performance (see Table 5).

The automatically generated queries were of varying quality in terms of relevance to their corresponding narrations (see Table 6). The queries did not turn out very diverse, most queries were generated on the form ”What is the person doing with ...?”. The prompt to LLM Gemma was changed asking it to generate more diverse queries, but it only slightly improved the output. Despite this, the results did increase after pretraining the model on these new queries, and hence the automatic generation of new queries worked sufficiently well. A possible extension of this investigation could be to use a different, more advanced, LLM to generate the queries to see if it gives an even better result.

An interesting finding during the work with this extension was that the model achieved higher performance at epoch 10 when it was trained for 15 epochs in total, than when it was only trained for 10 epochs (see Table 8). This probably has to do with the learning rate schedule that was used. Since the learning rate decreases linearly from the set learning rate to zero over the steps remaining after the warm-up phase, the learning rate is higher at epoch 10 when the model is trained for 15 epochs than when the 10th epoch is the last one. In this case one can see that it resulted in faster and more accurate learning up until epoch 10, since the performance of the model became higher. This finding indicates that the model could achieve higher performance if a hyperparameter search had been done. Doing a hyperparameter tuning was excluded from the scope of this project but could be an interesting further development of the project. The results from training the model for 15 epochs was included in Table 8 as an interesting addition, but since the model was trained for only 10 epochs in all previous parts of this project, those results were kept as the main results in Table 5 and Table 7. The time was not enough to rerun all previous steps with 15 epochs instead of 10 epochs.

5.3. Conclusion

This project investigated how to improve NLVL in egocentric videos using the Ego4D dataset. By comparing VSLBase and VSLNet architectures with different combinations of text and video feature encoders, the study found that VSLNet combined with BERT and EgoVLP features provided the best performance. As an extension, automatically generated queries from a language model were used to pretrain VSLNet, resulting in a significant boost of performance. These findings highlight the value of combining domain-specific visual features, powerful text encoders, and additional synthetic data for pretraining. Further improvement may be achieved through hyperparameter tuning and using more advanced query generation techniques.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021. 2

[2] Ryan Burke. GloVe, ELMo & BERT, 2021. Accessed 2025-06-23. `https://towardsdatascience.com/glove-elmo-bert-9dbbc9226934/`. 6

[3] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. 2

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 2, 3, 6

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2

[6] 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. 2, 3, 7

[7] 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. 1, 2, 3, 4, 5, 6

[8] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z Xu, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *Advances in Neural Information Processing Systems*, 35:7575–7586, 2022. 2, 3, 6, 7

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 3, 6

[10] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. 3

[11] 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. 1, 2, 3, 4, 5, 6