Guided Querying over Videos using Autocompletion Suggestions

Hojin Yoo
yoo.515@osu.edu
The Ohio State University
Columbus, Ohio, USA

Arnab Nandi
nandi.9@osu.edu
The Ohio State University
Columbus, Ohio, USA

ABSTRACT

A critical challenge with querying video data is that the user is often unaware of the contents of the video, its structure, and the exact terminology to use in the query. While these problems exist in exploratory querying settings over traditional structured data, these problems are exacerbated for video data, where the information is sourced from human-annotated metadata or from computer vision models running over the video. In the absence of any guidance, the human is at a loss for where to begin the query session, or how to construct the query. Here, autocompletion-based user interfaces have become a popular and pervasive approach to interactive, keystroke-level query guidance. To guide the user through the query construction process, we develop methods that combine Vision Language Models and Large Language Models for generating query suggestions that are amenable to autocompletion-based user interfaces. Through quantitative assessments over real-world datasets, we demonstrate that our approach provides a meaningful benefit to query construction for video queries.

ACM Reference Format:
Hojin Yoo and Arnab Nandi. 2024. Guided Querying over Videos using Autocompletion Suggestions. In *Workshop on Human-In-the-Loop Data Analytics (HILDA 24), June 14, 2024, Santiago, AA, Chile.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3665939.3665964

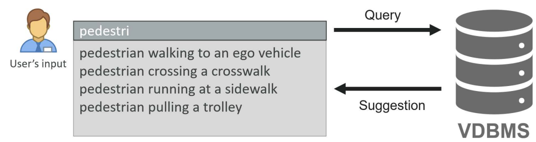
1 INTRODUCTION

The rapid rise in video content has motivated the need to manage, process, and effectively query such data. The proliferation of video data, driven by its increasing use in applications such as social media, IoT devices, and vehicle cameras underscores the urgency for more sophisticated data management solutions. Video content is not only large in volume but also complex in structure, making it difficult to manage and retrieve efficiently. While the database community has made several excellent strides in the area of video database management systems (VDBMS) [23, 49], unlocking a variety of impressive capabilities in analytical query processing [2, 6, 21], storage [11, 18], manipulation [46] and more. In this realm, we observe that the *specification* of queries poses an interesting *chicken-and-egg* problem: given the latent structure, complex content, and unstandardized terminology to refer to content in the videos, a user would have to watch some or all of the videos first, *before* they can specify a fully-informed VDBMS query. However, such a time investment in initially browsing videos may obviate the query processing benefits of a VDBMS. One approach to breaking this dependency cycle would be to implement systems that suggest queries based on contextual data and content analysis, thereby assisting users in formulating precise and effective queries without the need for extensive preliminary video review. However, this approach requires zero-shot [30, 52] automatic analysis and description of video content at a high enough quality level that it compares with a human-level description of video content. Capability of such levels has only recently been realized through very recent advancements in computer vision and Vision Language Models (VLMs) [29, 32]. These technologies now enable sophisticated analysis and zero-shot understanding of video data, opening new avenues for interactive and responsive query systems. When combined with the generative capabilities of modern Large Language Models (LLMs) [5, 45], we find a new opportunity to automate the creation of query suggestions for video data, which can be effective and useful for a user exploring a large collection of videos. Hence, our problem can be stated as: *Given a collection of videos, guide the user in a way that best assists in specifying relevant queries over video data,* where relevance is evaluated by the effectiveness of query suggestions in reducing the manual review needed and the accuracy of the retrieved video content. To address this problem, we look into the use of *autocompletion*, which provides a rapid, interactive, and iterative approach to query specification. At each keystroke, the user is presented with options to expand, enhance, and refine their queries. Users can either pick from these suggestions, or continue unassisted to further express their query. By gleaning insights about the contents of the video data from these suggestions, the user can learn about the exact contents of the video *while* they are typing out their query. This guided interaction loop of *query intent*$\circlearrowleft$*autocompletion suggestion*$\longrightarrow$*query result* is nearly imperceptible to the user, but dramatically improves the user's querying experience, as evidenced by its large-scale adoption in mainstream web search engines, email clients, and other software.

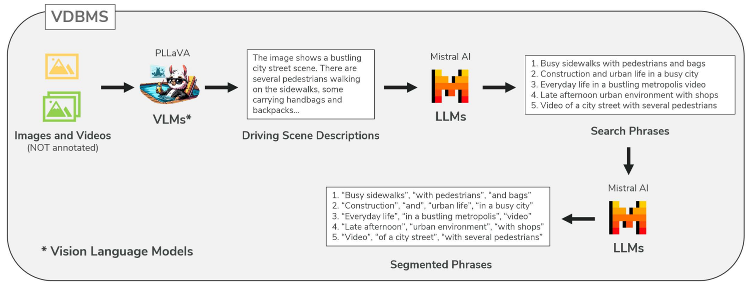
Alignment and Contributions: Our work aligns with the HILDA workshop's goals by enhancing the efficacy and usability of video database management systems through interactive autocompletion suggestions. By leveraging VLMs and LLMs to facilitate query formulation, it introduces innovative, human-in-the-loop approaches to exploring complex video data. Our contributions are:

HILDA 24 , June 14, 2024, Santiago, AA, Chile

ACM ISBN 979-8-4007-0693-6/24/06
https://doi.org/10.1145/3665939.3665964



(a) Query autocomplete scenario



(b) Procedure of generating search phrases

Figure 1: Overview of query suggestion system

- We provide a system that autocompletes queries for retrieval of video content based on zero-shot video understanding capabilities inherent in both VLMs and LLMs. This innovation streamlines the discovery process and enhances user interactivity with the multifaceted information extracted from video data.
- We leverage LLMs to simplify the search process by intelligently segmenting search phrases, minimizing the user's input effort for search completion.
- We demonstrate that our proposed method of creating a word graph for autocompletion significantly reduces the user's effort by measuring the number of keystrokes a user inputs to complete a query.

2 RELATED WORK

We build on a diverse body of work towards our vision for a video search query suggestion system, ranging from VLMs which integrate language and vision models, computer vision, and structured query autocompletion.

Vision Language Models: Recent advances in computer vision and natural language processing have led to the emergence of VLMs that combine vision and language models. VLMs can leverage the complementary strengths of both modalities to perform various tasks, making them a key research topic in multimodal artificial intelligence. CLIP [40], a pioneer in this space, achieves high zero-shot classification performance by jointly training on vision and language. ImageBind [15] demonstrated the potential of multimodal learning by training a single model to perform six cross-modal understanding tasks.

The integration of vision and language through LLMs [5, 20, 45] has been a focal point in recent research. Notably, Flamingo [1] and BLIP-2 [27] make significant strides in aligning the cross-modality of LLMs using web-scale image-text datasets. InstructBLIP [10] and miniGPT-4 [51] further push the boundaries of this integration by harnessing the power of pre-trained models to excel in vision-instruction tasks. These advancements highlight the potential of LLMs to make substantial contributions to the VLMs field. For instance, LLaVA [32] extends Vicuna [9] with a simple linear model with learnable parameters, enabling it to process text and images in the same space.

Video-LLaMA [50] and VideoChat [28] attempt to represent videos as embedding vectors using BLIP-2 [27] to enhance the understanding of videos through LLMs. Concurrently, Video-ChatGPT [33] proposes spatial and temporal pooling for video features. Additionally, LLaMA-VID [29] and PLLaVA [48] introduce a VLMs capable of encoding each video frame, allowing for the representation of longer videos.

Video Analytics using Computer Vision: There is a significant body of work focused on analyzing videos for tasks such as detection [19, 42, 43], tracking [4, 47], and estimating 3D locations [14, 41]. NoScope [22] is a system proposed for efficiently analyzing videos collected from stationary cameras. Instead of analyzing every frame using a video vision model, it utilizes an inference-optimized model to extract and analyze highlight frames. BlazeIt [21] is a method that reduces processing time by defining queries and user-defined functions (UDFs) for analysis, allowing only the information desired by the user to be processed. OTIF [3] propose a method to speed up processing using a segmentation proxy model to determine which frames need processing, thereby reducing unnecessary tasks. ExSample [36] suggests a method that involves scoring each sample. This strategy enables the analysis of specific instances in a video using a restricted set of samples. Monodepth2 [16] sought to analyze videos by estimating per-pixel depth from monocular videos. EVA [49] utilized exploratory video analytics by identifying and reusing the results of expensive user-defined functions. Spatialyze [26] is an end-to-end system that provides geospatial information about objects within videos using an object detector, 3D location estimator, and object tracker.

Present video analytics technologies are primarily designed for perception tasks, such as identifying objects within predefined categories. However, these technologies have their constraints. For example, object detectors are limited to recognizing only those objects they have been trained on, and the spatial data they provide is restricted to coordinates, which narrows down the scope of how a video's attributes are represented. However, by leveraging VLMs, we can offer users a unique and powerful tool. VLMs integrate the visual features of a video with linguistic descriptions, enabling multidimensional analysis beyond simple object recognition. This novel approach not only promises a deeper understanding of video content but also provides contextually relevant information in a way that was previously unexplored.

Guiding Structured Querying using Autocompletion: Autocompletion has become a pervasive input assistance mechanism, allowing users to not only reduce input effort [38], but also guide [39] the user to their intended queries in an interactive manner. Autocompletion mechanisms have been developed to go beyond just plain text, and work with rich schema [37], have tolerance towards errors [8] and awareness towards query contexts [25]. It has also served an excellent delivery mechanism for query recommendation [35] and personalization [7] based on a user's prior query histories. Video retrieval in response to semantic queries [30, 31, 44] is a foundational challenge in the field of video search, one that has been repeatedly highlighted for its significance. At the heart of this challenge is the imperative to operate in a zero-shot [12] capacity. The criticality of this approach is underscored by the dynamic nature of video content, which demands a system's ability to analyze and adapt to ever-changing scenarios. Given the unpredictable and varied landscape of video data, it is essential for such a system to possess the finesse to interpret and retrieve relevant videos accurately, without the need for prior training on specific cases. The availability of generative methods, VLMs and LLMs have now unlocked the ability for us to build on this body of prior work and extend into the area of video queries.

3 AUTOCOMPLETION FOR VIDEO SEARCH QUERIES

We introduce an innovative autocompletion system for video search queries designed to facilitate users in formulating queries by leveraging the automatic analysis of video data, as shown in Figure 1(a). This system not only suggests relevant search terms to users who may not be familiar with the video content in the database but also enhances the search experience by intuitively predicting user needs. While we expect our system to extend to more complex queries such as aggregation and data manipulation queries, for the scope of this paper we restrict ourselves to search queries. Detailed discussions in Section 3.1 articulate the use of VLMs for analyzing video scenes, while Section 3.2 elaborates on the algorithmic generation of search phrases for the autocomplete feature. Section 3.3 further explains how these search phrases are parsed, ensuring they are contextually appropriate and contribute to a more efficient and user-friendly video search process.

3.1 Scene Understanding via VLMs

An immense volume of video data is generated through the internet, mobile devices, and mobilities, accumulating in databases without specific categorization. VDBMS is a system that evaluates these videos and delivers the findings to users. With the advancement of deep learning, it has become possible to automatically analyze videos, such as tracking a suspect vehicle after examining CCTV footage. Consequently, users can input queries to extract desired information from vast video data. However, it's crucial to note that traditional VDBMS have limitations, as they can only utilize the information that deep learning models can provide. For instance, if a user wants to find a video of *"a man unloading a truck"* within the database, conventional VDBMS may be unable to perform such a task. This is a clear indication of the need for more advanced solutions, as traditional VDBMS are only capable of simple analyses, such as identifying the location and direction of objects in a video, but not complex analyses involving behaviors or states.

Multimodal Large Language Models (MLLM) are emerging as a solution to overcome these limitations. MLLMs are trained on data from cross-modality with different dimensions, allowing them to express the relationships between the characteristics of each modality. Integrating MLLMs with VDBMS makes it feasible to analyze video data and provide it to users. This system enables users to access desired videos, as exemplified in the scenario mentioned above. Therefore, we suggest integrating VLMs, a type of MLLM, with VDBMS. This integration aims to examine videos and utilizes the insights gained to refine and automatically complete user search queries.

3.2 Making Search Phrases for Autocompletion

Our system's approach to video data analysis leverages VLMs for their exceptional zero-shot scene understanding, enabling diverse user interactions that traditional VDBMS systems cannot facilitate. These VLMs can effortlessly interpret complex scenarios, such as a pedestrian crossing a crosswalk with a bag at an intersection, a truck navigating a bridge over a river, or vehicles halted at a red light. Integrated into the initial phase of our process, as illustrated in Figure 1(b), VLMs generate descriptive text for the videos. This text forms the foundation for search phrases, which are then stored in a database to inform query suggestions based on user inputs.

Following the initial phase, our system progresses to the next stage, where LLMs are instrumental in refining the search phrases. This is a crucial step in addressing the limitations of VLMs in text generation. Our empirical observations have shown that while VLMs excel at scene analysis, they can be less efficient at producing a varied textual output, particularly when tasked with generating multiple search phrases. For example, the following phrases were generated by the VLMs from a single video: *1. Traffic congestion, 2. City street, 3. Vehicle backlog, 4. Highway traffic, 5. Vehicle gridlock, 6. Roadway congestion, 7. Vehicle traffic jam, 8. City traffic, 9. Vehicle traffic, 10. Road congestion, 11. Vehicle traffic delay, 12. City traffic congestion, 13. Vehicle traffic slowdown, 14. Roadway traffic, 15. Vehicle traffic slowdowns, 16. City traffic congestion, 17. Vehicle traffic slowdowns, 18. Road congestion, 19. Vehicle traffic slowdowns, 20. City traffic.* This pattern of phrase generation by the VLMs is noteworthy; it does not aim to enumerate phrases for every discernible object within the video. Instead, it strategically focuses on a singular, salient feature, from which it systematically derives a series of related phrases. To overcome this, LLMs are introduced to produce a broader range of search phrases, thereby ensuring a more dynamic and effective query suggestion system. This innovative approach holds the promise of not just improving, but revolutionizing video search and analysis, making it more efficient and user-friendly.

3.3 Segmented Search Phrases

Our system's segmentation of search phrases is designed with the user's convenience in mind. By allowing for more semantic query fragments, we enhance the user experience. Building upon the search phrases generated in Section 3.2, our system formulates Directed Word Graphs (DWG) to facilitate the autocomplete function. Traditionally, each search phrase would be segmented into individual words to construct the DWG, leading to a recommendation of semantically incomplete information at the word level. For example, if the search phrase *"white car on a spacious road"* exists and a user types "white car", the next suggested word would be "on", followed by "a", which could be perceived as unnecessary steps in the query completion process.

To refine this, we have developed a system that segments search phrases not at the word level but into semantically meaningful sub-phrases before constructing the DWG. This approach parses the LLMs-generated search phrases into significant parts, as shown in Figure 1(b), to reduce unnecessary user input. Consequently, our system offers a more interactive video search query autocomplete experience by providing users with more contextually relevant and complete search phrase suggestions. This enables users to locate their desired content expeditiously with minimal exertion. Illustrative examples of the prompts utilized for generating search phrases within our system are depicted in Figure 2.

4 EXPERIMENT

For the evaluation of our system, we focus on the capability of LLMs to generate phrases for autocomplete purposes automatically. This section presents metrics that demonstrate how LLMs' generated and semantically parsed phrases impact user autocomplete experiences. To illustrate this, we employ a Minimal Keystrokes (MKS) metric used by Duan and Hsu [13] and Kharitonov et al. [24], which measures the number of key inputs a user requires to complete a query. We first introduce the methodology used to measure MKS in Section 4.1 and discuss the dataset utilized in Section 4.2. We describe the experimental setup in Section 4.3 and the experimental results in Section 4.4, followed by an outcomes analysis.

4.1 Evaluation Metric: Minimal Keystrokes

MKS is considered a characterization of the effort involved in entering a query while interacting with completions. It is defined as the minimum number of keystrokes required to achieve the target string. To demonstrate this, we construct the DWG from search phrases generated by LLMs for autocomplete. Let us consider that each search term, denoted as q , belongs to a set of queries Q, that a user intends to finish. Given that the user types in one character sequentially to create a set of partial query $P = \{p_i\}_{i=1}^{|P|}$, if a part of the completed query $\hat{q} \in q$ is within the top-k autocompletion suggestions when the user inputs p_i, then the query is completed as $\hat{q}$. The user continues to generate the query. However, if the desired part of the query is not present within the top-k autocomplete suggestions, leading the user to proceed with the next partial query. The value of MKS increases by 1. For instance, assume that a user interacts with the system to complete the *"white van driving on a larger road"* search query. Initially, the user types "wh" and the system suggests "white van". Next, the user enters "d" and the system recommends "driving". Finally, upon typing "on a l", the system completes the phrase with "on a larger road". In this sequence, the total MKS, counting all spaces, is 11.

As shown in Algorithm 1, we calculate MKS for all search phrases and then determine the average. We compare this metric for search phrases generated and semantically parsed by LLMs, calculating the average for each to assess how much our proposed method reduces user effort compared to the baseline. This demonstrates the effectiveness of predictive text systems in streamlining user input and enhancing overall search efficiency.

Algorithm 1 Pseudo code for Minimal Keystrokes

Input: Query set Q
Parameters: Length of autocomplete results k, Query q, Number of query trials t, Partial query set P
Output: Minimal key strokes set S

for q **in** Q **do**
    $P \leftarrow MakePartialQueries(q)$
    $t \leftarrow 0$
    repeat for each: $p_i \in P$
        $t \leftarrow t + 1$
        // Get k length of autocomplete results
        $searchResults \leftarrow SearchAutocomplete(p_i, k)$
        for $result$ **in** $searchResults$ **do**
            // If $result$ is part of completed phrase q
            if $result$ **in** q **then**
                Skip to $p_{i+|result|-|inp|}$
            end if
        end for
        // If no matched autocomplete, use next partial query
    until i < |q|
    $AddItem(S, t)$
end for

4.2 Datasets

For our experiments, we utilized the Driving Risk Assessment Mechanism with A captioning module (DRAMA) [34] dataset, which is captured from a moving vehicle on highly interactive urban traffic scenes in Tokyo. The dataset includes 17,785 scenario clips, each 2 seconds long, providing high-resolution footage synchronized with vehicle dynamics data. These clips were selectively filtered to highlight the driver's reactive behaviors to external stimuli that necessitate braking. The dataset is enriched with annotations across diverse dimensions, including video and object-level Q/A, risk assessments, and free-form captions. It features 17,066 risk scenarios with various vehicles, pedestrians, cyclists, and infrastructural elements. The dataset's free-form descriptive elements consist of 992 unique words, appearing over 306,000 times, offering a rich linguistic resource for analysis.

4.3 Experimental Setup

We randomly selected 1,000 videos from the DRAMA [34] dataset to generate text-based driving scenario, utilizing PLLaVA 13B [48] in the process. Based on the driving scenario, we created 20 search phrases for each video using LLMs, specifically employing Mistral-7B-Instruct-v0.2 [20]. These search phrases serve as a baseline in our approach, where each phrase is segmented into words to construct a DWG for performing autocomplete. Furthermore, we requested

Question / Prompt:
Describe all of things from the current driving scene.

PLLaVA Answer:
The image shows a bustling city street scene. There are several pedestrians walking on the sidewalks, some carrying handbags and backpacks. The architecture suggests a modern urban environment with a mix of buildings, including what appears to be a construction site with scaffolding. The street is lined with shops and restaurants, as indicated by the signage and storefronts. The sky is overcast, and the lighting suggests it might be late afternoon or early evening. There are no vehicles in motion on the street, and the overall atmosphere is one of everyday urban life.

Input video:

(a) Example of generated risk scenario from PLLaVA 13B [48]

Question / Prompt:
Give me 20 bullet points all the short search phrases I can ask to find the video with following scene description

--
[Answer from PLLaVA]

Mistral-7B-Instruct Answer:
1. Modern city street video without cars
2. Late afternoon urban hustle and bustle scene
3. Evening street with construction site
4. Busy sidewalks with pedestrians and bags
5. Scaffolding and shops in a mixed-use city
6. Video of a modern city street scene at dusk
7. Construction and urban life in a busy city
8. Everyday life in a bustling metropolis video
9. Late afternoon urban environment with shops
10. Video of a city street with several pedestrians
...

(b) Example of generated search phrases

Question / Prompt:
Split the following search phrase into each according to the semantic context as shown in example
Example: 'Cyclist in a narrow street environment' -> 'Cyclist', 'in a narrow street', 'environment'

--
[Answer from previous question]

Mistral-7B-Instruct Answer:
1. 'Modern city street', 'video', 'without cars'
2. 'Late afternoon', 'urban', 'hustle and bustle scene'
3. 'Evening', 'street', 'with', 'construction site'
4. 'Busy sidewalks', 'with pedestrians', 'and bags'
5. 'Scaffolding', 'and', 'shops' 'in a mixed-use city'
6. 'Video', 'of', 'a modern city', 'street scene', 'at dusk'
7. 'Construction', 'and', 'urban life', 'in a busy city'
8. 'Everyday life', 'in a bustling metropolis', 'video'
9. 'Late afternoon', 'urban environment', 'with shops'
...

(c) Example of segmented search phrases

Figure 2: Example prompt to make search phrases

semantic parsing of these search phrases, grouping semantically related parts to form a DWG, thereby developing an autocomplete system that leverages this structure.

We conducted experiments with the number of query suggestions, denoted as k, set at 1, 3, 5, and 10, to determine the MKS for each scenario. To generate scene descriptions using PLLaVA 13B [48], we input 16 frames per video, with the temperature set at 1.0. The creation of search phrases and the semantic parsing of these phrases were conducted using Mistral-7B-Instruct-v0.2 [20]. To implement the autocomplete system, we utilized the Flask [17] package in Python, operating in an environment powered by AMD EPYC 7643 processor.

4.4 Experimental Results

We explore the performance of our proposed system in enhancing the convenience of users' video searches. We investigate how the average MKS changes with varying values of k, and demonstrate the effectiveness of our method by presenting a boxplot of MKS when k is set to 10. The experiment was conducted on all phrases generated by LLMs.

Initially, we calculate the average MKS by varying k values at 1, 3, 5, and 10 for both the baseline and our proposed method of segmented phrases, as shown in Table 1. The baseline MKS decreases as the value of k increases, indicating that a higher number of autocomplete suggestions enables users to input less. When using segmented phrases, the MKS at $k = 1$ is similar to the baseline;

Table 1: Average Minimal Keystrokes

Method	k=1	k=3	k=5	k=10
Baseline	39.43	17.57	17.85	16.78
Segmented	36.85	9.79	8.21	6.78

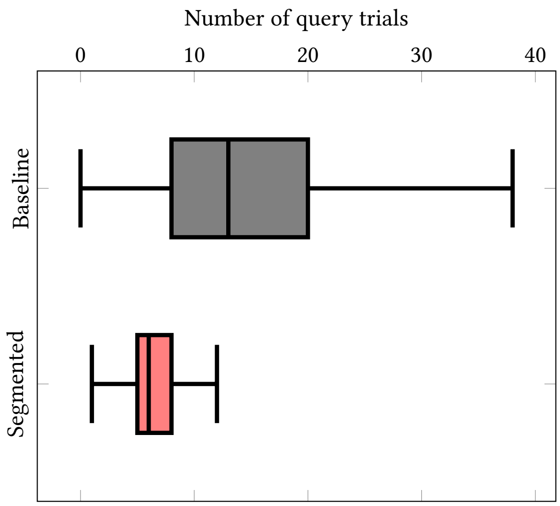


Figure 3: A distribution of Minimal Keystrokes when $k = 10$

however, as k increases, not only does the reduction in MKS become more pronounced compared to the baseline, but it is also notable that at $k = 3$ with segmented phrases, the MKS is even lower than the baseline at $k = 10$, underscoring the enhanced efficiency of our method. This suggests that parsing phrases with LLMs improves the quality of the user experience in video searches. Overall, our system, when suggesting up to 10 queries based on user input, requires on average **10 fewer inputs** to complete the desired search query compared to the baseline. The efficiency of the proposed method's queries can also be seen in the distribution of MKS for $k = 10$, as illustrated in the boxplot of Figure 3. These results show an increase in query suggestion performance, with a lower average and variance in MKS for query completion compared to the baseline.

4.5 Limitations

Bias in Phrase Generation Our research has identified a notable limitation concerning the search phrases produced by LLMs. The model prioritize the primary objects within a video, which can lead to significant content being overlooked. This bias in phrase generation often results in the description of only a fraction of the objects present—typically 1-2 out of 4-5. It is crucial to overcome this selective focus during phrase creation to ensure that users receive untruncated video search suggestions that capture the full scope of the content. Enhancing the algorithm to recognize and include secondary elements and background details could provide a more balanced and inclusive representation of the video's narrative, thereby enriching the user experience with more detailed and informative search results.

Limited Computational Resources: The limitations of our system are primarily related to its computational resource demands. Since our system generates search phrases based on the outcomes of VLMs, it requires a significant amount of computational resources. Unlike conventional VDBMS that could operate within a computational environment equipped with GPUs capable of running vision models, our system necessitates a High-Performance Computer (HPC) setting. Our environment also operated on servers with high GPU memory (NVIDIA A100) to run VLMs and LLMs. This requirement for substantial computational power diminishes the system's expandability, underscoring the urgent need for more lightweight VLMs to enhance it.

5 CONCLUSION AND FUTURE WORK

We present our initial vision for a system that is designed with the user in mind, guiding them to interact with videos and text using VLMs and LLMs. This is embodied as an autocompletion feature for a video search query system. By enabling users to articulate better video search queries, we aim to accelerate the adoption of video analytics across various fields. Our system demonstrates the capability to analyze unannotated visual data automatically and connect users with the desired content. Furthermore, by post-processing search phrases with LLMs, we show that it is possible to enhance the user experience of autocomplete functionality without analyzing users' search logs. In the future, we aim to develop a more precise video search capability by analyzing semantic video elements at the patch level, allowing us to articulate very fine-grained search predicates. In future work, we plan to conduct a user study to evaluate the effectiveness of video search suggestions utilizing VLMs in enhancing users' understanding of video content. Additionally, we intend to create or modify an existing VDBMS that leverages features from diverse data types, including visual, textual, geospatial, and depth information, to analyze mobility and travel patterns. This approach holds promise for advancing the field of multimodal interaction and making significant contributions to the data management domain.

REFERENCES

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems* 35 (2022), 23716–23736.
[2] Jaeho Bang, Gaurav Tarlok Kakkar, Pramod Chunduri, Subrata Mitra, and Joy Arulraj. 2023. Seiden: Revisiting Query Processing in Video Database Systems. *Proceedings of the VLDB Endowment* 16, 9 (2023), 2289–2301.
[3] Favyen Bastani and Samuel Madden. 2022. OTIF: Efficient Tracker Pre-processing over Large Video Datasets.(2022). (2022).
[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*. IEEE, 3464–3468.
[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
[6] Jiashen Cao, Karan Sarkar, Ramyad Hadidi, Joy Arulraj, and Hyesoon Kim. 2022. Figo: Fine-grained query optimization in video analytics. In *Proceedings of the 2022 International Conference on Management of Data*. 559–572.
[7] Gloria Chatzopoulou, Magdalini Eirinaki, Suju Koshy, Sarika Mittal, Neoklis Polyzotis, and Jothi Swarubini Vindhiya Varman. 2011. The QueRIE system for Personalized Query Recommendations. *IEEE Data Eng. Bull.* 34, 2 (2011), 55–60.
[8] Surajit Chaudhuri and Raghav Kaushik. 2009. Extending autocompletion to tolerate errors. In *Proceedings of the 2009 ACM SIGMOD International Conference*

on Management of data. 707–718.

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[10] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems* 36 (2024).

[11] Maureen Daum, Brandon Haynes, Dong He, Amrita Mazumdar, and Magdalena Balazinska. 2021. TASM: A tile-based storage manager for video analytics. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1775–1786.

[12] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9346–9355.

[13] Huizhong Duan and Bo-June Hsu. 2011. Online spelling correction for query completion. In *Proceedings of the 20th international conference on World wide web*. 117–126.

[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. 2018. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2002–2011.

[15] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15180–15190.

[16] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. 2019. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3828–3838.

[17] Miguel Grinberg. 2018. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.".

[18] Brandon Haynes, Maureen Daum, Dong He, Amrita Mazumdar, Magdalena Balazinska, Alvin Cheung, and Luis Ceze. 2021. Vss: A storage system for video analytics. In *Proceedings of the 2021 International Conference on Management of Data*. 685–696.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[20] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[21] Daniel Kang, Peter Bailis, and Matei Zaharia. 2018. Blazeit: Optimizing declarative aggregation and limit queries for neural network-based video analytics. *arXiv preprint arXiv:1805.01046* (2018).

[22] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. Noscope: optimizing neural network queries over video at scale. *arXiv preprint arXiv:1703.02529* (2017).

[23] Daniel Kang, Francisco Romero, Peter D Bailis, Christos Kozyrakis, and Matei Zaharia. 2022. VIVA: An End-to-End System for Interactive Video Analytics.. In *CIDR*.

[24] Eugene Kharitonov, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. 2013. User model-based metrics for offline query suggestion evaluation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 633–642.

[25] Nodira Khoussainova, YongChul Kwon, Magdalena Balazinska, and Dan Suciu. 2010. SnipSuggest: Context-aware autocompletion for SQL. *Proceedings of the VLDB Endowment* 4, 1 (2010), 22–33.

[26] Chanwut Kittivorawong, Yongming Ge, Yousef Helal, and Alvin Cheung. 2023. Spatialyze: A Geospatial Video Analytics System with Spatial-Aware Optimizations. *arXiv preprint arXiv:2308.03276* (2023).

[27] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[28] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355* (2023).

[29] Yanwei Li, Chengyao Wang, and Jiaya Jia. 2023. LLaMA-VID: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043* (2023).

[30] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. 2019. Zero-shot object detection with textual descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8690–8697.

[31] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. 2014. Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2657–2664.

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems* 36 (2024).

[33] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424* (2023).

[34] Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. 2023. DRAMA: Joint Risk Localization and Captioning in Driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1043–1052.

[35] Christopher Miles. 2011. More Like This: Query Recommendation for SQL. *Department of Computer Science & Engineering. University of Washington, Seattle, WA, USA* (2011).

[36] Oscar Moll, Favyen Bastani, Sam Madden, Mike Stonebraker, Vijay Gadepally, and Tim Kraska. 2022. Exsample: Efficient searches on video repositories through adaptive sampling. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2956–2968.

[37] Arnab Nandi and HV Jagadish. 2007. Assisted querying using instant-response interfaces. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. 1156–1158.

[38] Arnab Nandi and HV Jagadish. 2007. Effective phrase prediction. In *Proceedings of the 33rd international conference on Very large data bases*. 219–230.

[39] Arnab Nandi and HV Jagadish. 2011. Guided interaction: Rethinking the query-result paradigm. *Proceedings of the VLDB Endowment* 4, 12 (2011), 1466–1469.

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[41] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. 2021. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*. 12179–12188.

[42] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7263–7271.

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* 39, 6 (2016), 1137–1149.

[44] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. 2018. Find and focus: Retrieve and localize video events with natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 200–216.

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[46] Dominik Winecki and Arnab Nandi. 2024. V2V: Efficiently Synthesizing Video Results for Video Queries. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*.

[47] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, 3645–3649.

[48] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. PLLaVA: Parameter-free LLaVA Extension from Images to Videos for Video Dense Captioning. *arXiv preprint arXiv:2404.16994* (2024).

[49] Zhuangdi Xu, Gaurav Tarlok Kakkar, Joy Arulraj, and Umakishore Ramachandran. 2022. EVA: A symbolic approach to accelerating exploratory video analytics with materialized views. In *Proceedings of the 2022 International Conference on Management of Data*. 602–616.

[50] Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858* (2023).

[51] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).

[52] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. 2019. Zero shot detection. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 4 (2019), 998–1010.