Reinforcement Learning Enhanced PicHunter for Interactive Search

Zhixin Ma[1], Jiaxin Wu[2], Weixiong Loo[1], and Chong-Wah Ngo[1]

[1] School of Computing and Information Systems
Singapore Management University
Singapore
zxma.2020@phdcs.smu.edu.sg, {wxloo, cwngo}@smu.edu.sg
[2] Department of Computer Science, City University of Hong Kong
Hong Kong, China
jiaxin.wu@my.cityu.edu.hk

Abstract. With the tremendous increase in video data size, search performance could be impacted significantly. Specifically, in an interactive system, a real-time system allows a user to browse, search and refine a query. Without a speedy system quickly, the main ingredient to engage a user to stay focused, an interactive system becomes less effective even with a sophisticated deep learning system. This paper addresses this challenge by leveraging approximate search, Bayesian inference, and reinforcement learning. For approximate search, we apply a hierarchical navigable small world, which is an efficient approximate nearest neighbor search algorithm. To quickly prune the search scope, we integrate PicHunter, one of the most popular engines in Video Browser Showdown, with reinforcement learning. The integration enhances PicHunter with the ability of systematic planning. Specifically, PicHunter performs a Bayesian update with a greedy strategy to select a small number of candidates for display. With reinforcement learning, the greedy strategy is replaced with a policy network that learns to select candidates that will result in the minimum number of user iterations, which is analytically defined by a reward function. With these improvements, the interactive system only searches a subset of video datasets relevant to a query while being able to quickly perform Bayesian updates with systematic planning to recommend the most probable candidates that can potentially lead to minimum iteration rounds.

Keywords: Reinforcement Learning · Bayesian Method · Relevance Feedback · Interactive Video Retrieval

1 Introduction

As response time is critically important for a user-in-the-loop system, we focus on improving speed efficiency in terms of search space pruning and candidate selection for user feedback. The former is based on the state-of-the-art approximate near-neighbor search [10], which only traverses a small set of candidates for search result ranking. The latter is based on reinforcement learning to train a

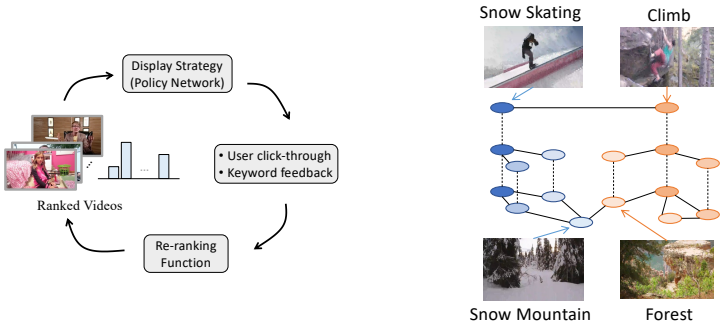

Fig. 1. Improvements over VIREO search system [9]: PicHunter++ (left), hierarchical navigable small world where more similar color shows higher similarity (right).

policy network that can recommend candidates to minimize user iterations. Intuitively, search space pruning addresses the issue of real-time system response. In contrast, the candidate selection strategy addresses the issue of mental tiredness by assisting a user to reach the search targets with minimum possible browsing efforts.

As most of the improvements are based on the integration of off-the-shelf techniques, this paper presents mainly our findings on PicHunter [2], which was proposed more than twenty years ago but remained one of the most competitive baselines [6,3,5] in video interactive search. For example, SOM-Hunter [4], a variant of PicHunter with a self-organizing map (SOM), has demonstrated strong performance in Video Browser Showdown (VBS). Empirically, we show that, even for the current VBS benchmark dataset with 1 million clips [1], PicHunter can achieve recall@1 $> 50\%$ with an average of 5 user iterations for 1 million visual known item queries under the ideal scenario that will be discussed later. Nevertheless, in practice, such high performance is not attainable due to the fact that human perception of visual similarity is different from what is defined in an ideal scenario. Instead of modeling human perception, we apply reinforcement learning to replace the greedy update in PicHunter with the goal of reaching the search target as soon as possible. This modification results in PicHunter recommending the candidates likely to speed up the search and browsing process rather than the candidates with the highest Bayesian scores. Figure 1 illustrates the two improvements made over the search system presented in VBS 2022 [9].

2 PicHunter

PicHunter [2] is an interactive video search system using relevance feedback. In each round of user iterations, the display strategy recommends user a set of video clips for selection. Based on the feedback, PicHunter uses Bayesian rules to re-rank the candidates and predict the search target. Specifically, let $D = \{d\}_{i=1}^{|D|}$ denote the set of video clips displayed to a user. The user will select video clips D^+, which are visually closer to the search target, and the remaining clips D^- are assumed irrelevant. Given the user's action, the system applies Bayesian rules to update the underlying probability distribution $P = \{p_i\}$, where p_i denotes the probability of the i-th candidate being the search target. To test the performance of PicHunter on visual known item search, we designed a user simulator to

provide feedback to the recommended candidate clips. It is worth mentioning that the search is purely based on visual similarity and does not involve textual query. The probability $P = \{p_i\}$ is initialized as 1, indicating that all clips are equally likely to be the search target. The system displays $|D|$ video clips in each turn for a user to select $|D^+|$ clips, which are more similar to the search target than the remaining $|D| - |D^+|$ clips. Both the selected and unselected clip sets will be used to update the probability distribution P as follows

$$p_i' = p_i \cdot \prod_{l \in D^+} \frac{exp^{\frac{-\delta_{cos}(v_f, v_l)}{\sigma}}}{\sum_{x \in D^- \cup \{l\}} exp^{\frac{-\delta_{cos}(v_u, v_l)}{\sigma}}} \tag{1}$$

where σ is a hyper-parameter to control the temperature scaling.

We conduct the experiments on two datasets, MSR-VTT [14] and V3C1 [1], to examine the effectiveness of PicHunter for visual known-item search (VKIS). There are 7,000 video clips in MSR-VTT and 100,000 clips in V3C1. In each iteration, the system will display a set of video clips D with the highest probabilities in P to a user. We employ a user simulator in the simulated experiments to provide relevance feedback. The simulator imitates an ideal situation where a user can always select the more similar clips from D by following the visual perception of the machine. Specifically, the machine vision of similarity is defined based on the cosine similarity of CLIP4Clip [7] features. The $|D^+|$ selected clips are more similar to the search target than the $|D| - |D^+|$ clips based on this machine vision. In the experiment, we set the display size, i.e., $|D| = 8$. We investigate the search performance in terms of recall@$\{1, 5\}$ by allowing a simulator to select $|D^+| < |D|$ "like number" of similar clips.

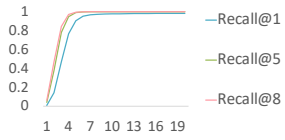

Fig. 2. The search performance (y-axis) over different rounds of iteration (x-axis) on MSR-VTT.

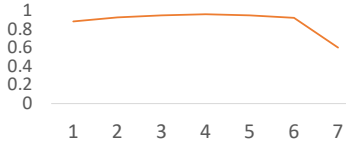

Fig. 3. Recall@1 (y-axis) versus like number (x-axis) within a maximum of 7 iteration rounds on MSR-VTT.

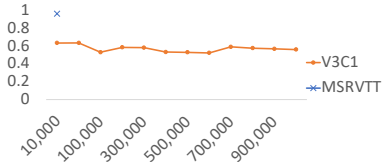

Fig. 4. Scalability test of PicHunter: recall@1 (y-axis) versus the number of clips (x-axis) within 7 rounds of user iterations. Note that the number of queries is set to 10,000.

Figure 2 shows the search performance averaged over 7,000 queries on the MSR-VTT dataset. The like number is set to $|D^+| = 4$. As noted, the recall performance increased sharply to more than 80% and saturated after 5 and 7 iterations for recall@5 and recall@1, respectively. To investigate the optimal number of clips to be feedbacked to PicHunter, Figure 3 shows the performance

trend of recall@1 by varying the like number. The result shows that the impact of likes is not significant. Even by selecting only one clip, the recall@1 can reach 88% within seven rounds of iterations. The performance gradually increases to 96% by providing four likes.

To access the scalability of PicHunter, we test the recall@1 performance by increasing the data size, as shown in Figure 4. Surprisingly, the performance trend does not vary proportional to the data size and stays around 53%-64% across different scales. On the V3C1 dataset with 1 million clips, recall@1 reaches around 56% within 7 iteration rounds with $|D^+| = 4$. Compared to MSR-VTT of 7,000 clips with recall@1=96%, the performance drop is considered significant. Nevertheless, the result shows that the performance depends more on the query difficulty than the data size. As observed, MSR-VTT queries are relatively easier to search compared to V3C1 with diverse visual content.

While PicHunter shows superior retrieval performance and resilience to data scale, the performance is highly dependent on whether the selected clips follow the perception of machine vision. We conduct another experiment to disrupt the ideal experimental setting by introducing noise to clip selection. Specifically, the simulator randomly selects one clip out of the four most similar clips to a search target in each round of user iteration. By doing this, recall@1 remains almost 0 on average after 7 iteration rounds. Basically, when the most similar clip, as perceived by the machine, is not selected, the Bayesian update cannot reflect the actual similarity distribution of the clips to a search target. In this situation, the recommended clips do not lead to the convergence of the search loop. In the experiment, even by randomly selecting three out of the four most similar clips as feedback, recall@1 barely reaches 20%.

3 PicHunter++

To remedy the limitations of PicHunter, we integrate the Bayesian update with reinforcement learning (RL). Specifically, the original display strategy in PicHunter, which greedily recommends the top $|D|$ clips most similar to a search target based on the probability distribution P, is replaced by RL. We train a policy network, a convolutional neural network, to take in the sorted probability P as input and output the probability $P^{'}$ as the distribution to sample $|D|$ clips for displaying. The network is optimized with the advantage actor-critic algorithm (A2C) with a reward function which is set to minimize the number of user iterations and maximize the recall performance. Unlike the greedy selection, which depends only on the historical update of P, RL provides a systematic mechanism for planning the navigation path to maximize future reward based on the current P. Furthermore, with RL, PicHunter can be more easily modified for textual known-item search (TKIS). For example, the policy network considers not only the liked clips selected by a user but also the textual query and user feedback interactively provided in different rounds for RL [8]. The probability distribution of clips, P, is still updated with the original Bayesian formula as in Equation 1.

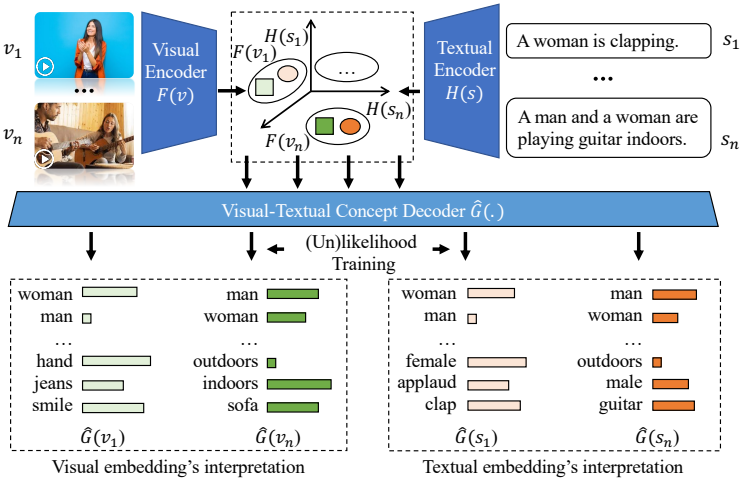

Fig. 5. The backend of VIREO engine which learns interpretable embeddings for both concept-free and concept-based searches.

4 Interactive Search System

Figure 5 shows the backend of VIREO search engine [13] where PicHunter++ operates on. Compared to our system in [9], the engine considers both likelihood and unlikelihood loss functions for learning interpretable embeddings. The engine produces an embedding of 2,048 dimensions for a video clip. For V3C1 [1], V3C2 [11] and the new-released marine dataset [12], such high-dimensional embeddings inhibit real-time interaction if linear search is performed on a typical laptop with 4-core CPU and memory size of 16G bytes. To improve system response time, we employ the state-of-the-art approximate k-nearest neighbor search, hierarchical navigable small world (HNSW) algorithm [10], to organize the embeddings as a hierarchical graph. As shown in Figure 1, the graph provides multi-layer indexing of embeddings for coarse-to-fine traversing of similar clips. The bottom layer indexes all the clips while the upper layers index only the representative clips. The connections from a representative clip to its children in the lower layer are linked (shown as dotted lines in Figure 5) for progressive search. Given a textual query, the search starts by comparing the query embedding with the embeddings of representative clips indexed at the top layer. The search progresses to the next layer by comparing only to the children of representative clips retained at the upper layer of the graph. The progressive traversal and search of candidates across layers avoid exhaustive comparison and hence significantly cut short the processing time.

5 Conclusion

We have presented the major improvements over our interactive system [9]. Specifically, the Bayesian inference from PicHunter [2] helps to narrow down the search space. The policy neural network improves the display strategy in PicHunter, boosts the recall performance and further reduces the interaction rounds. For efficient search, we employ HNSW algorithm to index the video

embeddings. We also improve the backend of the VIREO search engine using unlikelihood training.

Acknowledgment

This research was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant.

References

1. Berns, F., Rossetto, L., Schoeffmann, K., Beecks, C., Awad, G.: V3c1 dataset: An evaluation of content characteristics. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. p. 334–338. ICMR '19 (2019)
2. Cox, I.J., Miller, M.L., Minka, T.P., Papathomas, T.V., Yianilos, P.N.: The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. IEEE transactions on image processing **9 1**, 20–37 (2000)
3. Heller, S., Gsteiger, V., Bailer, W., Gurrin, C., ór Jónsson, B., Loko, J., Leibetseder, A., Mejzlík, F., Peka, L., Rossetto, L., Schall, K., Schoeffmann, K., Schuldt, H., Spiess, F., Tran, L.D., Vadicamo, L., Veselý, P., Vrochidis, S., Wu, J.: Interactive video retrieval evaluation at a distance: comparing sixteen interactive video search systems in a remote setting at the 10th video browser showdown. International Journal of Multimedia Information Retrieval **11**, 1 – 18 (2022)
4. Kratochvíl, M., Mejzlík, F., Veselý, P., Soucek, T., Loko, J.: Somhunter: Lightweight video search system with som-guided relevance feedback. Proceedings of the 28th ACM International Conference on Multimedia (2020)
5. Kratochvíl, M., Veselý, P., Mejzlík, F., Lokoč, J.: Som-hunter: Video browsing with relevance-to-som feedback loop. In: MultiMedia Modeling. pp. 790–795. Springer International Publishing, Cham (2020)
6. Loko, J., Veselý, P., Mejzlík, F., Kovalcík, G., Soucek, T., Rossetto, L., Schoeffmann, K., Bailer, W., Gurrin, C., Sauter, L., Song, J., Vrochidis, S., Wu, J., ór Jónsson, B.: Is the reign of interactive search eternal? findings from the video browser showdown 2020. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **17**, 1 – 26 (2021)
7. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: CLIP4Clip: An empirical study of clip for end to end video clip retrieval. arXiv:2104.08860 (2021)
8. Ma, Z., Ngo, C.W.: Interactive video corpus moment retrieval using reinforcement learning. p. 296–306. MM '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3503161.3548277
9. Ma, Z., Wu, J., Hou, Z., Ngo, C.W.: Reinforcement learning-based interactive video search. In: MultiMedia Modeling. pp. 549–555. Springer, Cham (2022)
10. Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**, 824–836 (2020)
11. Rossetto, L., Schoeffmann, K., Bernstein, A.: Insights on the v3c2 dataset. arXiv preprint arXiv:2105.01475 (2021)
12. Truong, Q.T., Vu, T.A., Ha, T.S., Lokoc, J., Tim, Y.H.W., Joneja, A., Yeung, S.K.: Marine video kit: A new marine video dataset for content-based analysis and retrieval. In: MultiMedia Modeling, MMM 2023 (2023)
13. Wu, J., Ngo, C.W., Chan, W.K., Hou, Z.: (un)likelihood training for interpretable embedding (2022). https://doi.org/10.48550/ARXIV.2207.00282
14. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5288–5296 (2016)