Video Browser Showdown by NUS

Jin Yuan, Huanbo Luan, Dejun Hou, Han Zhang,
Yan-Tao Zheng, Zheng-Jun Zha, and Tat-Seng Chua

School of Computing, National University of Singapore, Singapore
{yuanjin,zhazj,chuats}@comp.nus.edu.sg,
{luanhuanbo,houdejun214,zhanghan8788,yantaozheng}@gmail.com

Abstract. The known item search task (KIS) aims to retrieve a unique video or video clip in the video corpus. This paper presents a novel interactive video browsing system for KIS task. Our system integrates visual content-based, text-based and concept-based search approaches. It allows users to flexibly choose the search approaches. Moreover, two novel feedback schemes are employed: first, users can specify the temporal order in visual and conceptual inputs; second, users can label related samples with respect to visual, textual and conceptual features. Adopting these two feedback schemes greatly enhances search performance.

1 Introduction

Recently, a new video search task, named the "*Known Item Search*" (KIS), has been proposed to simulate a real-world video search scenario [1]. In KIS task, users aim to find a desired video or video clip that has been seen and known before by mining their memory. Compared to the typical video search task, finding relevant sample in KIS task is more difficult for two reasons: first, there is only one right answer for each user's query; and second, users may not be able to completely describe the content of the desired video or video clip according to their memory, especially when it happened a long time ago.

This paper presents a multifunctional and friendly video browsing system for KIS task. The system assembles the visual content-based, text-based and concept-based approaches. Users can flexibly select one or more approaches to search video results according to their memory. Moreover, our system supports the use of temporal sequence and related samples to enhance the search performance.

2 Video Browsing System

2.1 Framework

Often, a user only remembers a part of visual scenes or a part of textual information in the desired video. Therefore, it is necessary to provide a multi-modality search platform as shown in Figure 1. Users can input a visual, textual, or conceptual query. These different queries are respectively used by a visual model, textual model, or conceptual model to retrieve the results. The final search results are then generated by fusing the individual results from these models, where

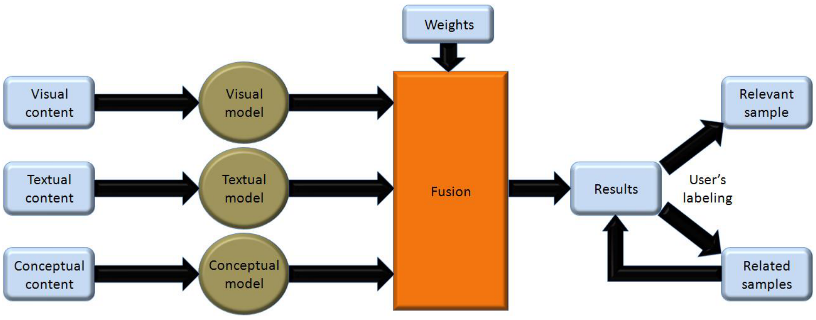

Fig. 1. The framework of our video browsing system.

the fusion weights are manually set by the users. Browsing the search results, users can label "*related samples*" [2] (the samples that are similar to the relevant samples), or renew queries to find the desired video.

2.2 Visual Content-based Search

When users remember some visual scenes of the desired video in their memory, they can draw visual content on a sketchpad to form a visual query Q^v. For each visual query Q^v and each keyframe K_i in the dataset, it is divided into multiple visual blocks, where each block B_j^v (B_j^i) is represented as a feature vector $f(B_j^v)$ ($f(B_j^i)$) based on Color feature. The visual matching score $R(Q^v, K_i)$ between Q^v and K_i is calculated based on cosine distance as:

$$R(Q^v, K_i) = \tfrac{1}{J} \sum_{j=1}^{J} Cos(f(B_j^v), f(B_j^i)) \qquad (1)$$

where J is the number of the blocks in Q^v drawn by the users.

To enhance search performance, users can draw multiple visual contents and specify the temporal order among them. This temporal relationship can be used to enhance the prediction accuracy. Let $\mathcal{Q} = \{Q_1^v, Q_2^v, \ldots, Q_M^v\}$ be the sequence of visual contents drawn by the users, where Q_m^v occurs before Q_{m+1}^v. The relevance score of K_i with respect to $\mathcal{Q}$ is calculated by considering two factors: the visual matching score of K_i to one of the visual contents ($R(Q_m^v, K_i)$ in Eq (2)); and the visual matching scores of its temporal neighboring keyframes to the other visual contents. Here, for the other $M-1$ visual contents, we select $M-1$ keyframes from the $2W$ neighboring keyframes to maximize the value of relevance score. Moreover, we require the temporal order of visual contents to be consistent with that of the selected keyframes (see the constraints in Eq (2)).

$$\begin{aligned} R(\mathcal{Q}, K_i) = & \max_{1 \le m \le M} \{\max_{p_b} \prod_{b=1}^{m-1} R(Q_b^v, K_{p_b}) \cdot R(Q_m^v, K_i) \cdot \max_{p_a} \prod_{a=m+1}^{M} R(Q_a^v, K_{p_a})\} \\ s.t. \quad & p_b \in \{i-W, i-W+1, \ldots, i-1\} \quad p_a \in \{i+1, i+2, \ldots, i+W\} \\ & p_1 < p_2 < \ldots < p_{m-1} < p_{m+1} < p_{m+2} < \ldots < p_M \end{aligned} \qquad (2)$$

2.3 Text-based Search

The system allows users to select the categories for the desired video. Through the user interface, user can specify whether the desired video clip contains speech or subtitle. In addition, users can also indicate the semantic category which the desired video clip belongs to. We define seven semantic categories $\{C_k\}_{k=1}^{7}$ ("*Music*", "*Entertainment*", "*Education*", "*Science*", "*Comedy*", "*News*", "*Cartoon*") suggested by the YouTube website. For each category C_k, we downloaded the tag files of the top 100 videos from YouTube. These tag files were merged and expressed as a normalized text vector T_k^c. Furthermore, for each video clip V_n in the dataset, we extracted textual words by ASR and OCR. After filtering stop words, V_n is represented as a normalized text vector T_n^v. Finally, we calculate a relevance score between V_n and C_k by considering two factors: The semantic closeness of V_n to C_k which is measured by the Google distance between T_n^v and the category name C_k; and the co-occurrence between the text words from V_n and C_k which is measured by the cosine distance between T_n^v and T_k^c. We balance these two factors with a weight parameter. When users select one category, the system will rank the results according to the relevance scores.

2.4 Concept-based Search

Users can express their query as a sequence of concept bundles [3], and specify the temporal order among them. For example, the query {("*car*", "*sky*"), ("*lady*", "*singing*")} describes the desired video containing at least two shots: one containing the concepts "*car*" and "*sky*", that occurs before the other one containing both "*lady*" and "*singing*". For each concept bundle in the query, we calculate a relevance score for each keyframe by the concept-based video search approach [3]. The relevance score of a keyframe to the query (a sequence of concept bundles) is calculated by the same approach as Eq (2).

2.5 Related Sample-based Search

While browsing the search results in the interface, users can label "*related samples*" with respect to three features (visual content, text, or concept). For example, user can label a keyframe as visually related sample if he thinks that it is visually similar with one of the keyframes in the correct video. To distinguish different kinds of related samples, we allocate three areas in the interface, and users can drag one kind of related samples from search results to the corresponding area. Furthermore, for each keyframe in the dataset, we previously found its K nearest samples according to the three features respectively. Once the users click a related sample, its K nearest samples are shown in the interface.

References

1. X. Y. Chen, J. Yuan and et.al. TRECVID 2010 Known-item Search by NUS. TRECVID Workshop, 2010.
2. J. Yuan, Z. -J. Zha and et.al. Utilizing Related Samples to Enhance Interactive Concept-based Video Search. IEEE Transactions on Multimedia, 2011.
3. J. Yuan, Z. -J. Zha and et.al. Learning Concept Bundles for Video Search with Complex Queries. Proc. of ACM Int. Conf. on Multimedia, 2011.