J. Vis. Commun. Image R. 20 (2009) 104–116

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Query representation by structured concept threads with application to interactive video retrieval ☆

Dong Wang [a,*], Zhikun Wang [a], Jianmin Li [a], Bo Zhang [a], Xirong Li [b,1]

[a] State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[b] Intelligent Systems Lab Amsterdam, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 2 June 2008
Accepted 11 December 2008
Available online 24 December 2008

Keywords:
Interactive video retrieval
Query concept mapping
Concept thread
Structured query formulation
Concept tf-idf
Semantic feedback
Query representation
TRECVID

ABSTRACT

In this paper, we provide a new formulation for video queries as structured combination of concept threads, contributing to the general *query-by-concept* paradigm. Occupying a low-dimensional region in the concept space, concept thread defines a ranked list of video documents ordered by their combined concept predictions. This localized representation incorporates the previous concept based formulation as a special case and extends the restricted AND concept combination logic to a two-level concept inference network. We apply this new formulation to interactive video retrieval and utilize abundant feedback information to mine the latent semantic concept threads for answering complex query semantics. Simulative experiments which are conducted on two years' TRECVID data sets with two sets of concept lexicons demonstrate the advantage of the proposed formulation. The proposed query formulation offers some 60% improvements over the simple browsing search baseline in nearly real time. It has clear advantages over *c-tf-idf* and achieves better results over the state-of-the-art online ordinal reranking approach. Meanwhile, it not only alleviates user's workload significantly but also is robust to user mislabeling errors.

1. Introduction

Since the revolution in web information retrieval, multimedia (video, especially) retrieval has been regarded as the next grand challenge for accessing and managing the huge amount of information involved. Until now, videos are mainly accessed from noisy text associated with the content, whether automatically recognized speech, closed captions, or social tags. Though standard text retrieval approach has been successfully applied in web search, the achievement of this *query-by-text* paradigm is limited in the video domain. The reason is that apart from its noisy nature, the succinct text usually does not elaborate on the visual obvious. A *query-by-image* paradigm, which aims at mapping low-level image features of color, texture and edge directly to arbitrary complex information need, has been limited by the *semantic gap* [1]. In a quest to narrow the semantic gap, a few hundreds of semantic concepts are defined and detected automatically quite recently: LSCOM [2] has defined a lexicon of 834 semantic concepts for news video, including various roles of people, objects, scenes and events; 363 concepts are detected in [3], 374 in [4] and 311 in [5], all from LSCOM and leveraging on generic learning approaches, though with varied performance below the user expectation. For each video clip[2], predictions are made to indicate the confidence of all concepts' presence and stored as semantic indices. These concept indices direct to a new *query-by-concept* video search paradigm for video achieve access. For example, a query as "scenes with snow" can be retrieved with concept *Snow*, but not likely with *Fire*.

In the *query-by-concept* paradigm, one needs to represent video query as the combination of selected concepts, and rank the video documents through their relevance to these concepts, often in the form of a weighted sum of concept prediction scores. The former step, also known as query-concept-mapping (QUCOM), is necessary since specific queries are relevant to only a few concepts. This representation implicitly assumes that relevant clips about a query gather at the top of the rank lists of all relevant concepts. However, given the limited size of the concept lexicon currently available, this assumption may not be valid due to the following two observations. First, the well matched target concepts can be out of the vocabulary (OOV). Often we run into some super-concept, or

☆ This work is supported by the National Natural Science Foundation of China under the Grant Nos. 60621062 and 60605003, the National Key Foundation R&D Projects under the Grant Nos. 2003CB317007, 2004CB318108 and 2007CB311003, and the Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

* Corresponding author.
E-mail address: dwang97@gmail.com (D. Wang).

[1] Work performed while the author was at Tsinghua University.

[2] Usually videos are segmented at the shot granularity, defined as one uninterrupted camera taken.

1047-3203/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2008.12.001

hypernym, of the missing one. For example, when concept *Basketball* is missing, we may leverage on *Sports*. When hypernym is used, it may be problematic to assume that relevant video clips about the specific concept gather at the top of rank list of its hypernym. See Fig. 1 for a real-world example. Second, even with in-vocabulary concepts, information needs often add certain restrictions on the relevant concepts. It is quite possible that clips containing the restricted concept are not always clustered at the top of the corresponding rank list (cf. Section 5.2 for empirical support). Subsequently, representing the query as direct combination of concept scores may not be optimal. In contrast, it may be desirable to identify the region in the concept prediction score space which is most densely populated by the target OOV/restricted concept. These observations are rather general since the problems still persist when the number of concepts grows to 5000, or even 10,000.

However, given the brief text query description, together with a few query images sometimes, it is difficult to identify relevant concepts in automatic video retrieval (AVR) [6,16,17], not to say mining relevant OOV concepts. In interactive video retrieval (IVR), users are often required to browse a large amount of data before finding results satisfying the information need. Thus current day IVR systems put great emphasis on interface design to present search result efficiently, e.g. [7–9], where the *query-by-concept* paradigm serves as one important component. Nonetheless, as Christel [10] suggests, in spite of its great potential for future video retrieval, *query-by-concept* risks a too complex interface as the number of concepts grows from tens to one thousand. More importantly, users cannot search with OOV concepts. Thus, it seems difficult to achieve satisfactory *query-by-concept* by designing novel user interfaces alone. As an alternative, a more expressive query representation which relaxes the top-rank relevance assumption may produce improved retrieval experience. Unlike in the automatic mode, the significant amount of interactive user feedback may provide necessary information for such purpose.

Thus motivated, we propose a new concept-thread based formulation for describing video query in this paper and apply it to interactive video retrieval. Partitioning the whole concept index score range into some sub-ranges, we obtain many concept threads as a certain low-dimensional region in the concept prediction score space, like *Sports*: [0.65,0.70] in Fig. 1 as one example. The region is called as thread since it contains a list of examples ranked by their respective concept prediction score combinations. Viewing the feedback examples as points in the semantic concept space, we try to cover the relevant points with a small number of threads. This feedback by semantic thread formulation (abbreviated as SemanFeed hereafter) naturally takes concept performance into account. Threaded concept is a localized query representation which permits dynamically updated thread following. It also extends the simple AND concept combination logic to a two-level concept inference network with possible diverse logic types. Concept thread is a departure from the concept based query representation by allowing flexible combination scheme and query logic for video retrieval.

Given a query, SemanFeed generates a few candidate concepts, and generate subsequent single-concept threads by dividing the candidate concepts into regions, and then adopts an Apriori [11] like algorithm to mine possible concept combinations to form additional multi-concept threads. See Fig. 2 for an illustrative example with query "multiple people in formation", where one multi-concept thread from *Crowd* and *Protesters* is presented in the first query component. As new feedback data arrive, threads are iteratively formed, the structured query representation is updated and a list of ranked documents is produced considering both thread relevance and concept relevance.

To validate the proposed SemanFeed approach, we conduct simulative experiments on the TRECVID 2005 and 2006 data sets with two different lexicons of 311(Tsinghua)/374(Columbia) concept detection results. The results demonstrate that the proposed query formulation offers some 60% improvements over the simple browsing search baseline in nearly real time. SemanFeed has clear advantage over the simple yet powerful *c-tf-idf* approach and, while being more efficient, achieves better result over the state-of-the-art online ordinal reranking [12] approach. Experiments also confirm that SemanFeed not only alleviates user's workload significantly but also is robust to user mislabeling errors.

We organize the remainder of this paper as follows. We introduce related works in Section 2, present the proposed query representation in detail in Section 3, and apply SemanFeed to interactive retrieval in Section 4. Then we present the experimental results in Section 5 and conclude this paper in Section 6.

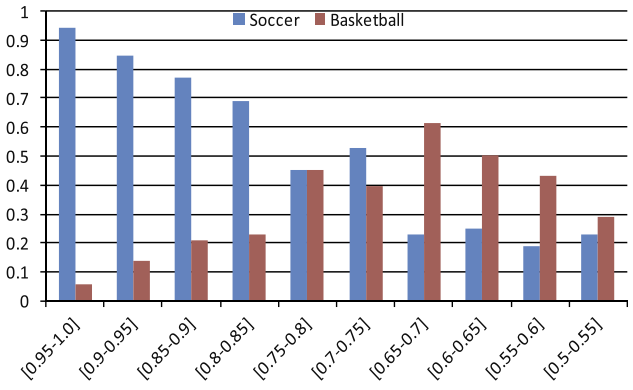


Fig. 1. The statistic of video clips about *Soccer* and *Basketball* in the *Sports* index on TRECVID 2006 data set. Ranges on *x*-axis denote *Sports* prediction score range and numbers on *y*-axis the proportion of *Soccer* and *Basketball* in each range. Video clips about *Basketball* densely populated in [0.5–0.8] while those about *Soccer* in [0.75–1].

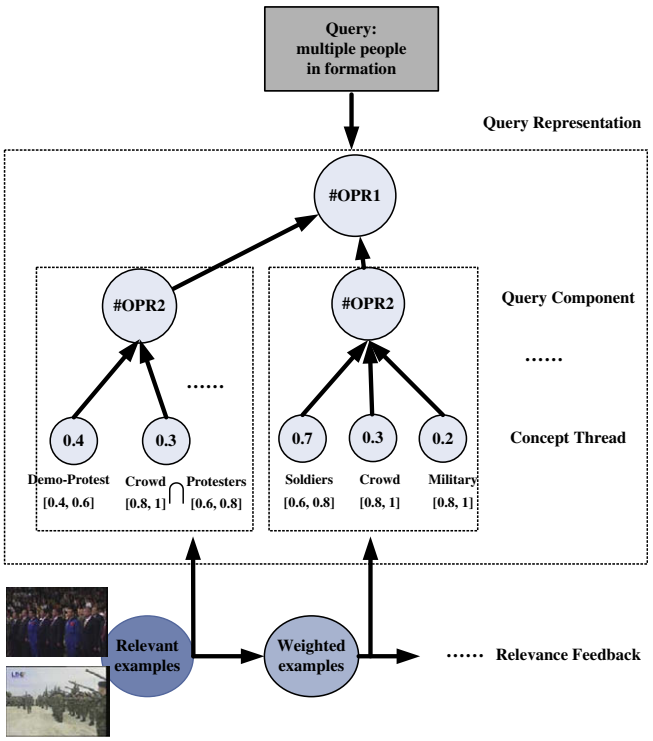


Fig. 2. The query representation as structured concept-threads.

2. Related work

The generic concept detection problem has been treated extensively in the past few years [13–15] while the research of using semantic index for retrieval get more attention only recently. A few studies positively support the usefulness of the *query-by-concept* paradigm for video retrieval [3–5,16–19]. However, some of these results are too optimistic in using oracle selected concepts [18] or human judged relevant concepts [19], and only the top concept is selected in [3]. Others use a rather small lexicon (<40) of concepts [16,17,19] where no sufficient concepts can be selected. Therefore they bypass the QUCOM problem, and subsequently the query representation problem.

A comparison of the standard text retrieval and query-by-concept video retrieval paradigm helps us better understand why QUCOM is indispensable for concept-based query representation. Viewing shots (the basic video retrieval units) as visual documents and concepts as visual terms, the parallelism between text and video documents can be created. In the former paradigm, the query and the documents reside in the same feature space, and a few query words are explicitly given. However, in the latter paradigm, the query keywords, at best, only implicitly specify the semantic concepts, in another different feature space. By solving QUCOM, a few relevant concepts are selected and subsequent retrieval is performed in the semantic concept space. In a recent review paper [20], Natsev *et al.* name the QUCOM problem as semantic concept-based query expansion and explicitly categorize QUCOM in automatic retrieval by the information involved as three kinds of text based, visual based and retrieval result based QUCOM. We adopt this categorization in our review. We further identify an important categorization dimension as the granularity of the query representation because through finer granularity, the problem of the limited concept lexicon can be alleviated to answer complex queries.

2.1. QUCOM in automatic video retrieval

It is a fast solution to link query to concepts by text matching between the query text and concept description [4] or a predefined concept ontology [3] if explicit semantic relation can be established between the query and concept descriptions [6,21]. Neo et al. [17] further take the concept detection performance into account, with the same text match approach. However, this line of research ignores the visual aspect of the concepts, which might be also important for solving QUCOM. For example, it is not straightforward to relate *Mosques* to the query "helicopters in flight" or relate *Kitchen* to "people reading a newspaper" by direct text match. However, they are really relevant as the connection can be mined through visual cues [5]. To count for the semantic correlation among concepts in QUCOM, an ontology-enriched semantic space is proposed [22] to enable modeling and reasoning concepts in a linear space.

Query images, if provided, establish visual links between user information need and semantic concepts. Predicting concepts on one query image, the resulting scores can be concatenated as a vector in the concept space. Searching by this full vector without performing QUCOM resembles a very verbose text query with all words in the lexicon present, as presented in previous work [16,23,24]. The irrelevant concepts will not only decrease the response time significantly, but also severely degrade the retrieval performance [5]. Treating concepts as basic "visual terms" for describing visual documents, we propose a *tf-idf* like scheme in our previous work [5]. The *c-tf-idf* scheme combines both concept popularity and concept specificity for a given query. In the search process, the well-established vector space model or language model can be deployed. Experimental results [5] show that *c-tf-idf* is among the best approaches for QUCOM in AVR. Despite its robustness, *c-tf-idf* does not consider possible correlation across concepts and the concept index performance variations. It also restricts itself to a simple AND logic for concept combination to express the complex query semantics. For example, in searching "multiple people in formation", the most salient concept combination from LSCOM may be comprised of *Demonstration_Or_Protest* and *Crowd*. However, another combination of *Soldiers*, and *Military* may provide novel relevant results. The two sets of concepts should be combined in an OR logic instead of the AND logic taken by *c-tf-idf*.

The result based QUCOM takes inspiration from the pseudo-relevance feedback approach. It examines initial retrieved documents for a query topic as pseudo-relevant/irrelevant examples in order to select discriminative concepts to improve the retrieval performance. It holds a strong assumption that the initial results are of sufficient quality. Hsu *et al.* [25] first propose to rerank the video search results via the information bottleneck principle. A robust probabilistic local context analysis (pLCA) approach is adopted in [20]. Kennedy and Chang [26] mine the search results to discover and leverage concept co-occurrence patterns for reranking the initial search result. To the same end, Yang and Hsu [12] adopts an online ordinal reranking framework based on ListNet [27]. Ordinal reranking changes the optimizing goal from classification to ranking and the underlying ListNet algorithm solves the efficiency for the ranking problem involved.

Combination of textual and visual cues for better QUCOM [28,20] is also possible. Fusing *query-by-concept* with *query-by-text* and *query-by-image* often brings further improvements [5,29,20]. However, given the limited supervision information available, it is hard for these approaches in AVR to fully discover the potential of the *query-by-concept* paradigm and overcome the shortcomings of limited lexicon size, high index noise and simple AND combination logic, not to say further exploit finer granularity query representation.

2.2. QUCOM in interactive video retrieval

Compared with AVR, the IVR mode enjoys the benefit of user judgements for superior performance, often 200% gain in percentage, as witnessed by past TRECVID [30] campaigns. Current day IVR systems put more emphasis on user interface design rather than on query analysis, e.g. [7–9]. One typical example is the Extreme Video Retrieval (XVR) [8], which delivers one RSVP (Rapid Serial Visual Presentation) or MPVP (Manual Paging with Variable Pagesize) display and requires user's extensive effort to browse significant portion of examples to find more relevant items. Beside XVR, several novel interfaces are designed by the MediaMill group [9] as *Galaxy Broswer* and *Cross Broswer*, *Rotor Broswer* and more recently *Fork Broswer*. All these browsers emphasize the multi-modal nature of the video retrieval process by allowing users to visualize significant data portion in multiple dimensions. A collaborative search scheme [31] is recently designed to exploit the synergy of multiple users based on the current, active search behavior of one's fellow searchers. Among many other studies with *expert* users who are familiar with the retrieval system, a recent user study [10] explores the use of the search system by government intelligence analysts and reports a usage of the *query-by-concept* paradigm 36% of the search time on average. Noticing that this community is dominated by text search system expertise, their better performance on and favor for a system with concept shows the practical utility of the *query-by-concept* paradigm. In designing a static browsing unit for news video corpus, de Rooij et al. [32] describe various forms of related video fragments as video threads, such as textual similarity, image feature similarity or temporal adjacency. Semantic threads are also built by consider the concept space as a whole and run *k*-means clustering using some similarity

function. Clearly, they do not use the same word thread for concept-based query representation. Rather they are doing document clustering instead.

Given the immature video retrieval technology, the synergy between user and retrieval system should be explored. With a predefined large-scale concept lexicon, the user feedback, no matter explicitly or implicitly given, provides valuable supervision information to further exploit the latent power of the *query-by-concept* paradigm. As a natural result, the user's mental load can be greatly relieved since browsing with highly relevant concepts may increase the chance of finding relevant shots. Few studies are conducted to explore the concept space for more efficient feedback. In our previous work [29], we apply *c-tf-idf* to IVR and significant improvement over the text baseline is achieved. However, *c-tf-idf* is not expressive enough to fully explore the feedback information because after a few feedback examples are given, the weights for the selected concepts will become nearly constant.

To the best of our knowledge, almost all formulation for *query-by-concept*, no matter in AVR/IVR, are based on the concept level instead of finer thread granularity. The only exception is our previous work [33] where a concept thread (called as concept segments in [33]) feedback mechanism is proposed. However, it aims not at a new formulation for video query and does not provide the important understanding of why thread based representation is better over concept based representation. In this paper, we provide a general thread based representation for video query and fully develop the SemanFeed approach, addressing issues related to improved parameter estimation and multi-concept threads. In addition, extensive experiments are also carried out, including empirical query ground-truth analysis, sensitivity analysis to varied concept detection results and initial rank list, and comparative study with some state-of-the-art reranking method.

3. Representing query with semantic concepts

In this section, after presenting the preliminaries and the general retrieval model, we briefly introduce the concept based query representation as a basis. Then we present the structured thread based query representation and the corresponding retrieval model.

3.1. Preliminaries

Let $L = \{c_j\}$ be a lexicon of concepts with $M = |L|$ as the number of concepts. Let $C = \{s_i\}$ be a video corpus and $N = |C|$ is the number of documents in C. Each shot (visual document) s_i is represented by a concept vector $\mathbf{d}_i = [d_{i1}, d_{i2}, \ldots, d_{iM}]^T \in \mathcal{V}$, $\mathcal{V}$ being the concept space $\boldsymbol{d}_i$ resides. Note that $d_{ij} = P(c_j|s_i)$ is the probabilistic prediction score of concept c_j occurring in s_i. Let the scalar form d_i also denote shot s_i. In interactive retrieval, the user feedback specifies a series of examples (shots) clicked as $F = \{s_{i_t}, y_{i_t}\}_{t=1}^{T}$ where i_t is the index of the shot chosen at time t, corresponding to a series of points $\{\mathbf{d}_{i_t}\} \subset \mathcal{V}$. Here $y_{i_t} \in \{-1, 1\}$ is the user provided label for irrelevant and relevant examples, respectively. Let F^+ (F^-) denote the set of those relevant (irrelevant) examples.

By viewing concepts as visual terms and shots as visual documents, the parallelism between video and text document is naturally created. With this premise, we leverage on the well-founded model in the information retrieval field, i.e., *Vector Space Model* to perform video retrieval [34]. It considers a document d^t and a user query q^t as M^t-dimensional vectors $\mathbf{d}_w^t$ and $\mathbf{q}_w^t$, respectively, where each dimension is a weight associated with a distinctive term and M^t is the size of the term lexicon. We denote the superscript t for text and add subscript w to emphasis that they are not raw term counts. The relevance of d^t with regard to q^t is measured as

$$R(d^t, q^t) := \mathbf{d}_w^t \cdot q_w^t = \sum_{i=1}^{M^t} w(i, d^t) w(i, q^t) = \sum_{q_i^t \in q_t} w(i, d^t) w(i, q^t), \quad (1)$$

where $w(i,d^t)$ and $w(i,q^t)$ are weights for the ith term. $w(i,d^t)$ measures the importance of the ith term for document d^t and $w(i,q^t)$ for query q^t similarly. Intuitively $R(d^t,q^t)$ measures the similarity between d^t and q^t and important terms for both document and query are emphasized. Often a query only contains a few terms q_i^t. This allows for efficient inverted list operations to speedup the retrieval process.

Unlike text retrieval where keywords are provided explicitly, user usually does not specify the relevant concepts for a video query. A video query q can be represented as some plain text with/without additional query images. An automatic retrieval model accepts the query input and generates an initial ranked list for interactive retrieval. During the feedback process, a QUCOM algorithm finds a subset of relevant concepts L_q, which corresponds to a few dimensions in $\mathcal{V}$. After getting L_q, the same vector space model ranks each shot d by its relevance to query q, defined as

$$R(d, q) := \sum_{c_j \in L_q} w(j, d) w(j, q), \quad (2)$$

where for each concept $c_j \in L_q$, two weights $w(j,d)$ and $w(j,q)$ are associated with the shot and the query, respectively. Similarly, the weights should consider both concept occurrence frequency in one shot and occurrence frequency across shots.

3.2. Concept based query representation

The central problem of concept based query representation is concept selection and weight assignment. Here we introduce the *concept tf-idf* (*c-tf-idf*) approach to show how concept based query representation works. *c-tf-idf* takes inspiration from the *tf-idf* term weighting scheme in the information retrieval field. By viewing concepts as virtual terms (the occurrence frequency of a concept in a shot is a real value in [0,1]), we obtain the *c-tf-idf* metric. The intuition is that highly probable concepts in the query are more likely to be relevant; concepts occur in too many documents might be less informative. The *c-tf-idf* of concept c_j in a shot d_i is defined as

$$c - tf - idf(c_j, d_i) = freq(c_j, d_i) \log\left(\frac{N}{freq(c_j)}\right), \quad c_j \in C, \quad (3)$$

where $freq(c_j,d_i)$, the occurrence frequency of c_j in d_i, is approximated by $d_{ij} = P(c_j|d_i)$, and $freq(c_j) = \sum_i freq(c_j, d_i)$ is the occurrence frequency of c_j in the corpus and approximated similarly. $P(c_j|d_i)$ is the probability of finding c_j in d_i, estimated by the concept detectors. The essence of this *tf-idf* based concept selection method is to pick out concepts which maximally reduce the uncertainty of the corpus's relevance to the query [35].

Thus we calculate both $w(c,d)$ and $w(c,q)$ in Eq. (2) by the *c-tf-idf* metric. The *c-tf-idf* calculation for a shot d is quite straightforward since both $freq(c,d)$ and $freq(c)$ are known. Given a query q with a few query image examples, *c-tf-idf* takes the averaged occurrence frequency of the image examples as the occurrence frequency estimation for each concept. When a certain feedback information F is collected, we assume F express the query need and generate $w(c,q)$ similarly with the averaged occurrence frequency. More specifically, we take $freq(c_j, q) = \frac{1}{|F^+|}\sum_{i \in F^+} d_{ij}$ and calculate $w(c,q)$ by Eq. (3). $w(c,q)$ measures the relevance of concept c to query q. So we select the top k concepts with highest query *c-tf-idf* score. Given the selected concepts, shots are ranked by relevance score defined in Eq. (2) for search and further fusion with other retrieval results.

c-tf-idf is a robust approach since it takes the average of concept prediction scores of multiple relevant examples, as shown in Eq. (3). However, when over tens of relevant examples are discovered in the interactive feedback process, the *c-tf-idf* metric becomes almost fixed and is too robust to incorporate more information.

Yang and Hsu [12] takes *c-tf-idf* as the concept selection method and iteratively re-estimates the weights for the selected concepts in an ordinal ranking framework. They assume that automatic retrieval result is meaningful, and adopt the ListNet [27] algorithm to align the combined concept scores with the retrieval result by tuning the weight for each selected concept.

3.3. Structured concept threads for query representation

Observing the fact that relevant shots for a query may not be top-ranked ones in the corresponding concept lists, we are motivated to relax the concept based formulation by introducing concept threads. To put it simple, query representation by structured concept threads can be divided into two parts of thread definition and mapping query to structural threads. Each concept index is a ranked list of the shots in the corpus. Threads for one concept can be defined as a partition of the concept index into some non-overlapping regions, better still if the partition has some explicit semantic interpretation. Also multi-concept threads are also possible since often specific concept can be identified by certain combination of two broad-meaning concepts. For example, *Athlete* is *People* who is playing *Sports*. Subsequently, *Basketball Player* may be implied as *People* [0.7, 0.8] and *Sports* [0.6, 0.7].

Given a non-overlapping partition function H_j:$[0,1] \rightarrow \{l\}$ for a concept c_j, a partition bins $\{b_{jl}\}$ of the concept index can be derived as $b_{jl} = \{d, H_j(d) = l\}$, satisfying $\cup_l b_{jl} = C$ and $b_{jl} \cap b_{jl'} = \varnothing$ for $l \neq l'$. The partition function can be a simple histogram partition, or an adaptive function with respect to the distribution of the scores in the concept index. The continuity of the partition function is required. Each b_{jl} serves as a concept thread candidate. The joint bin of b_{jl} and $b_{j'l'}$ of different concepts c_j and $c_{j'}$, defined as $b_{jlj'l'} = d, H_j(d) = l$ and $H_{j'} = l'$, may also provide additional multi-concept threads.

Given the single-concept threads, together with some metric to measure their utility to a query, a stronger combination of multi-concept threads can be mined by adopting the Apriori [11] algorithm. Take the two-concept thread as an example, it defines actually a region in the two-dimensional concept space and covers a few relevant examples. It is a stronger representation for possible AND logic among sub-concepts. By searching only the combination of threads with large p^+_{jl}, we can control the computational complexity at reasonable level.

Given the threads, an important problem of measuring thread's utility with respect to a query should be considered. If a large pool of past queries can be obtained, together with some click-through data, one can leverage on the implicit feedback information, and possibly additional semantic information to relate the current query to existing threads. Otherwise, explicit feedback information can provide such kind of cue, as we will explain in Section 4. A metric reflecting the strength of the thread b_{jl} for a query should be defined, from semantic correlation, query relatedness or the probability of relevant example occurrence in b_{jl} as $p^+_{jl} = P(y_i = 1 | d_i \in b_{jl})$. We choose this p^+_{jl}, relevant example ratio (*rel-ratio*) as the metric. A detailed description of estimating p^+_{jl} is given shortly afterwards in Section 4.1. However, one relevant example can belong to several bins of different concepts with different p^+_{jl}. Noticing that, simply presenting the thread with the largest p^+_{jl} to the user omits the impact of other bins and the respective *c-tf-idf* scores of each example. Instead, combining the concept level *c-tf-idf* score and the thread level p^+_{jl} estimation well balances the two cues. More formally, we adapt the vector space model as

$$R(d,q) = \sum_{c_j \in L_q} \sum_{l} p^+_{jl} w(c_j, d) w(c_j, q) \delta(d \in b_{jl}), \tag{4}$$

where $\delta(x)$ is the indicator function which is 1 when x is true and 0 otherwise. Given an inverted list of the concept indices, $\delta(x)$ can be implemented very efficiently. Similarly, multi-concept threads emphasize the documents in the joint bin of both concepts and their scores are defined as $p^+_{jlj'l'} \delta(d \in b_{jlj'l'})(w(c_j, d)\ \ w(c_j, q) + w(c_{j'}, d)\ w(c_{j'}, q))$. For the sake of understanding, we can take the threads as possible potential sub-concepts, and each with a probability to indicate its usefulness for the current query. In this way, the vector space model remains unchanged but our concept index changes. It will be interesting to see how to mine real sub-concepts from the query feedback information and we leave it as an important future work.

Until now, we are treating each query as one weighted sum of a certain concepts/concept threads. This corresponds to the simple weighted AND (WAND) logic of the underlying concept threads. However, in many cases, the query information need could be more complex. Take the query "multiple people in formation" as an example, the most salient concepts may be *Demonstration_Or_Protest*, *Crowd* and *Protesters*. However, another concept combination of *Soldiers*, *Crowd* and *Military_Personnel*, may also contribute some relevant results. We term the WAND combination of a few concept threads as the query component hereafter. An OR logic will be suitable to combine two WAND query components. Thus we propose a two-layer query inference network for the structured query representation, as shown in Fig. 2. Multiple query components can be generated again through various information source, e.g. from multiple related queries identified from the query pool. Each query component q_r defines a relevance score as in Eq. (4). The OR logic is implemented as a max operation of the scores obtained from multiple query components as

$$R(d,q) = \max_r R_r(d,q). \tag{5}$$

This two-layer query inference network can incorporate different kinds of query operations and it is possible to take different types of query operations with respect to a given query.

4. Mining implicit semantics from feedback analysis

4.1. Generating concept threads

Given the feedback information as points in the concept space, concept threads can be obtained by covering the points with low-dimensional regions. This is essentially a distribution density estimation problem with much noise present. Assuming a parametric form of the distribution, the Expectation Maximum (EM) algorithm can be applied. However, EM is slow and it is difficult to decide proper distribution form which fits for all different queries. Meanwhile, the relevant examples are found through a browsing process, and may possibly not be sampled according to its underlying distribution. Considering the huge amount of noise involved, we take the non-parametric histogram estimation as a robust partition scheme for the concept index. This solution is also a trade-off between accuracy and real-time update needs. For interactive retrieval, real-time execution is a must since users cannot tolerate over 2 s of delay while searching [36]. Even with this simplified representation, searching over all concepts in the lexicon is neither necessary nor feasible at feedback time. Since the *c-tf-idf* measure provides a rough estimation of the usefulness of the concept to a query, we first filter out a few concept candidates for generating the actual threads. We also fix the size of the histogram bins to be equally split and adaptively increase the bin size as the feedback examples accumulate. More specifically, we set

$|H| \propto \sqrt{|F^+|}$ as in standard non-parametric distribution estimation, together with minimal and maximum limits of $|H|_{\min} = 5$ and $|H|_{\max} = 20$, respectively. These bins serve as thread candidates.

Now we discuss how to estimate p_{jl}^+. Given the number of relevant examples in b_{jl} as $|F_{jl}^+|$ or more briefly as r_{jl}, and the browsed examples in b_{jl} as $|F_{jl}|$ or a_{jl}, a straightforward estimation would be $p_{jl}^+ = \frac{r_{jl}}{a_{jl}}$. A Laplacian smoothing of

$$p_{jl}^+ = \frac{r_{jl}+1}{a_{jl}+1} \tag{6}$$

prevents both zero division and omitting those threads where no relevant examples are discovered to that point. However, the number of total shots in b_{jl}, o_{jl}, can be counted in for estimating p_{jl}^+. One variation is

$$p_{jl}^+ = \frac{r_{jl}+1}{o_{jl}+1} \tag{7}$$

which simply replaces a_{jl} to o_{jl}. This is a robust and conservative estimator and degrades gracefully to the original *c-tf-idf* approach when only one histogram bin is used. A more complex Bayesian inference method assumes that the presence of the relevant/irrelevant examples in a given thread b_{jl} obeys the binomial distribution with a Beta conjugate distribution as $f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$. Here $B(\alpha,\beta)$ is a normalization constant, α and β are hyper-parameters. The posterior distribution can be derived as

$$p_{jl}^+ = \frac{r_{jl}+\alpha}{o_{jl}+\alpha+\beta}. \tag{8}$$

The hyper-parameter can be chosen in proportion to the total number of shots in b_{jl} as

$$\alpha = \frac{o_{jl}}{\sum_l o_{jl}} r_{jl}, \quad \beta = \frac{o_{jl}}{\sum_l o_{jl}} (a_{jl} - r_{jl}). \tag{9}$$

From another point of view, this update can be seen as a smoothing technique.

To further improve the estimation accuracy of *rel-ratio*, we use a kernel function to smooth the examples. So here we allow the example counts to be real number. We take the Epanechnikov kernel which has the profile

$$k_E(x) = \begin{cases} 1-x, & 0 \leqslant x \leqslant 1, \\ 0, & x > 1. \end{cases}$$

The Gaussian kernel could be an alternative choice. Accordingly, the bandwidth h which controls the spread of each example should be considered as well. As examples accumulate, shrink the bandwidth h accordingly and we will asymptotically arrive at the exact relevance estimation, as guaranteed by kernel density estimation literature, e.g. [37]. Also, when the concept indexing performance can be estimated before search, we can choose the bandwidth h for each concept independently with respect to its performance since lower performance corresponds larger score variance, which can be reflected in h. However, in the current datasets (as detailed in Section 5), we do not have reliable performance estimation and fix h to half of the histogram bin size.

Given single-concept threads, we can generate multi-concept threads as outlined in Section 3.3. In this study, we allow first half of the threads ordered by p_{jl}^+ to be searched and accept the combined thread only when the joint thread with $p_{jj'll'}^+ > 5p_{jl}^+ p_{j'l'}^+$. These multi-concept threads enrich the representation.

4.2. Structured query representation

Given the thread generation process, the two layer query inference network is produced sequentially by a boosting like greedy search method. In one round of feedback, first a few concept threads are obtained as one query component. Because this query component may only explain part of the examples, a weight update of the examples emphasizes the not well-explained ones. Thus, we select a few new threads as a second query component and this can be repeated until the desired number of query components (N_v) is obtained. Please see Fig. 3 for details of this process, where we define the weight output for each relevant example to account for the contribution of the all threads containing it in Step 2.b.

For each query component q_r with weighted examples, both *c-tf-idf* and p_{jl}^+ are upgraded to a weighted version. Accordingly, q_r defines a relevance score $R_r(d,q) = \sum_{c_j \in L_q^r} \sum_l p_{jl}^+ \delta(d \in b_{jl}) w(c_j,d) w(c_j,q)$ where L_q^r is the set of selected concepts for threads in round r. The OR logic is implemented as a max operation over multiple query components, as defined in Eq. (5).

4.3. Fusion via reranking

Query-by-concept, on its own, still cannot achieve optimal retrieval performance since other modalities (e.g., text) also provide useful ranking information. We study this multi-modal fusion

Given $F^+ = \{d_i^+ \in F \text{ where } y_i = 1\}$, N_v the number of query components.

(1) Initialize the weights $w_i = \frac{1}{|d_i^+|}$.

(2) For r = 1 to N_v:
   (a) Generate concept threads with examples F^+ and estimate p_b^+ with Eq. 8. Generate multi-concept threads accordingly.
   (b) Form query component q_r as the WAND operation of these threads.
   (c) Compute the weighted output of d_i^+ under q_r as $h(d_i^+) = \sum_{b \in q_r} p_b^+ \delta(d_i^+ \in b)$ where p_b^+ is the *rel-ratio* of b.
   (d) Update $w_i = w_i \exp^{-\gamma h(d_i^+)}$ and normalize so that $\sum_i w_i = 1$ where γ is a predefined constant.

(3) Output $\{q_r\}$.

Fig. 3. The structured query components generation process.

problem under a reranking framework. That is, given a search result list obtained from certain modalities (e.g. text), we target at improving the search quality by reranking the list with *query-by-concept* results. A linear fusion model, though simple, has been shown adequate to fuse visual and text modalities in video retrieval [25,12]. Given a search result list, we divide the reranking process into two steps: (1) obtain a reranked list from the *query-by-concept* modality; (2) linearly combining the initial list with the new reranked list, as

$$sim_{new}(d,q) = \beta \cdot sim_{initial}(d,q) + (1-\beta) sim_{rerank}(d,q), \quad (10)$$

where $\beta \in [0,1]$ is a weighting factor, indicating the bias on the two ranked lists. $\beta = 1$ means no reranking is introduced at all; $\beta = 0$ means totally reranked. Ideally, the list with higher precision should be more favored. We use an unbiased weighting scheme for the sake of simplicity, i.e., setting $\beta = 0.5$.

With regard to $sim(d,q)$, we use a robust rank-based normalization method [38],

$$sim(d_i,q) \approx \frac{N+1-i}{N}, \quad i = 1,\ldots,N, \quad (11)$$

where d_i is the ith shot in the ranked list, and N the list's length.

5. Experiments

After introducing the experimental setup, experiments are carried out in five parts: First of all, we show the necessity of QUCOM and the insufficiency of the concept based query representation through empirical evidence. Secondly, we present the detailed parameter settings of the proposed representation with the SemanFeed approach for feedback. Then comparative studies of SemanFeed, *c-tf-idf* and ordinal reranking [12] are conducted, together with comparison of two sets of concept detectors and two different initial rank lists in Section 5.4. Finally, SemanFeed's error tolerance and efficiency are reported.

5.1. Experimental setup

It has been always difficult to obtain enough data to evaluate multimedia systems due to copyright issues and the sheer volume of data required. Thanks to NIST for the TRECVID video retrieval benchmark, an open, metric-based evaluation via a common large data set for video retrieval and indexing techniques is thus available to the research community. So experiments are carried out on both TRECVID'05 (TV05) and TRECVID'06 (TV06) data set, following the standard benchmark and the official Average Precision (AP) measurement to ensure the comparability across system. TRECVID'07 dataset is not used because the data set mainly are educational programs while the LSCOM concept training set is original annotated on TRECVID'05 TV news programs. This train-test distribution mismatch violates the basic independently identical distribution assumption of the indexing algorithm and causes severe indexing performance degradation. Thus TRECVID'07 is not helpful to test the proposed representation.

5.1.1. Data sets and measurement

TV05 and TV06 share the same training set (TV05d) but have different test set (TV05t and TV06t). The whole set (TV05+06) contains 310-h multilingual news video captured from MSNBC/NBC/CNN (English), LBC/ALH (Arabic) and CCTV/PHOENIX/NTDTV (Chinese). The video data are divided in shots and each shot is represented as a few keyframes. An official set of ~300k image keyframes are extracted. Table 1 provides some statistics of the data sets.

We use 37 multimedia search queries defined in TV05 and TV06 for the experiments. The selected queries express diverse information needs concerning general people, things, events, locations, etc. and combinations of these needs. Please refer to [30] for more details on query topics and data sets. We excludes 11 "PersonX" query topics since those query topics rely on *query-by-text* paradigm strongly and do not conform to our goal.

The performance is evaluated by Average Precision (AP) on shot level. Given a ranked list L, AP is defined as $\frac{1}{R}\sum_{j=1}^{S}\frac{R_j}{j}I_j$ where R is the number of true relevant instances in a set of size S; R_j the number of relevant instances in the top j instances; $I_j = 1$ if the jth instance is relevant and 0 otherwise. It can be seen as an approximation to the area under the Precision–Recall curve. The relevant shots are judged by NIST using a pooling method. To compare results across queries, Mean Average Precision (MAP) is defined as the mean AP scores involved for all queries.

The concept models for indexing are trained on the leave-out TV05d data set and concept indices are generated by fusing the prediction results on the test set, respectively. TV05d is annotated with 449 concepts from the LSCOM [2] multimedia concept ontology, from which 311 concepts with more than 20 positive examples are chosen to index. The concept index generation process consists of three stages of multiple feature extraction, classifier training and fusion. We use five kinds of features, SVM classifier and a simple average fusion algorithm for the three respective stages. This lexicon is referred as concept311 hereafter. Its indexing mechanism is among the state-of-the-art approaches, as proven by past TRECVID benchmark. Please compare [5,15] for more details. In addition, another external set of concept lexicon, 374 concept detectors from Columbia University [39] (referred as concept374 hereafter), is also evaluated to test the sensitivity of SemanFeed to underlying concept detectors.

5.1.2. Evaluation protocol

User's behavior is one of most important factors influencing the interactive retrieval performance. To compare different approaches fairly and to facilitate the labeling effort, we use the officially provided ground truth by NIST as a surrogate to simulate the human labeling process. Analysis of previous user log shows that user can browse average 2000 shots during a 15-min process (standardized TRECVID search time). Since the temporal dimension is proven to be very effective among diverse functions of the browsing interface, we incorporate this simulative action also, as shown in the evaluation protocol outlined in Fig. 4.

5.2. Why c-tf-idf is not enough?

Though TRECVID queries are designed towards utilizing the smaller LSCOM-Lite lexicon of 39 concepts, a small number of concepts in a larger LSCOM lexicon also contribute to the overall search performance. To show that we exhaustively evaluate each concept index for each TRECVID'06 query in automatic retrieval

Table 1
Data set statistics.

	TV05d	TV05t	TV06t	TV05+06
Length (h)	80	80	150	310
Shots	44k	46k	80k	170k
Keyframes	75k	78k	144k	297k
Time span	October–November 2004	November–December 2004	November–December 2005	October–December 2004/2005

setting and count the number of concepts with $AP > 0.01$. The result is plotted in Fig. 5. Clearly, many queries have only a few (<10) relevant concepts, indicating the necessity to perform QUCOM. While on average 6.1 concepts are relevant to a query, a large standard deviation variance of 7.3 concepts is present across different queries. Also, almost no concept can help retrieval of "PersonX" queries. This observation demonstrates the importance of performing QUCOM when we intend to exploit the concept modality.

Another ideal experiment tries to show that even with relevant concepts, relevant examples may often clustered somewhere in the middle of the ranked concept index instead of only heavily populated at the top of the index, as illustrated in Fig. 6. In this experiment, the ground truth shots for four randomly selected queries are identified in the four hand-selected relevant concept indices. Their respective *rel-ratio* is calculated and plotted. Except query 163 ("meeting with a large table and more than two people") which is only densely populated at the top position, all three others have some other densely populated threads. For query 161, there exist no relevant results in the top ranked shots until the prediction confidence is around 0.8. Given the large number of relevant shots (1245 for 161 and 1160 for 163) and the state-of-the-art concept indexing technique adopted, the significance is evident. This experiment shows the necessity of using SemanFeed to mine the potential threads. One possible reason for this clustering phenomenon is the hypernym surrogate for a target concept. The specific nature of query need also supports our local clustering assumption, although clusters found in this way sometimes may lack clear semantic meaning.

5.3. Parameter settings

We evaluate the influence of different parameters here, including: the number of threads in one concept N_t, the number of concepts N_c in one query component, the number of query components N_v and the OR query logic. In these experiments, we fix the default parameters as $N_t = 5$, $N_c = 3$, $N_v = 3$ and $\gamma = 10$ unless otherwise stated.

Parameter N_t. We run an experiment with fixed N_t by allowing N_t varying in [1,15] and show the result in Fig. 7. Note that $N_t = 1$ is *c-tf-idf* with the re-weighting scheme. We find that $N_t = 5$ consistently yields the best performance on both TV05 and TV06. Then we adopt the adaptive bin size scheme which has a similar performance with that of $N_t = 5$. So fixing $N_t = 5$ is enough for the limited amount of feedback examples, at least for the current data set.

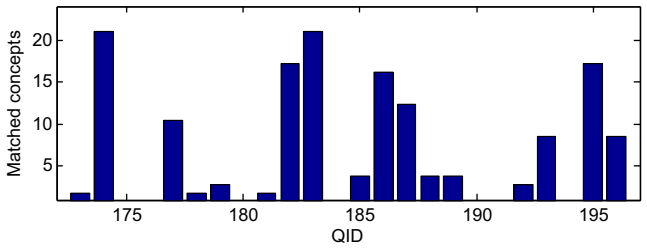


Fig. 5. The number of the relevant concepts for 24 queries.

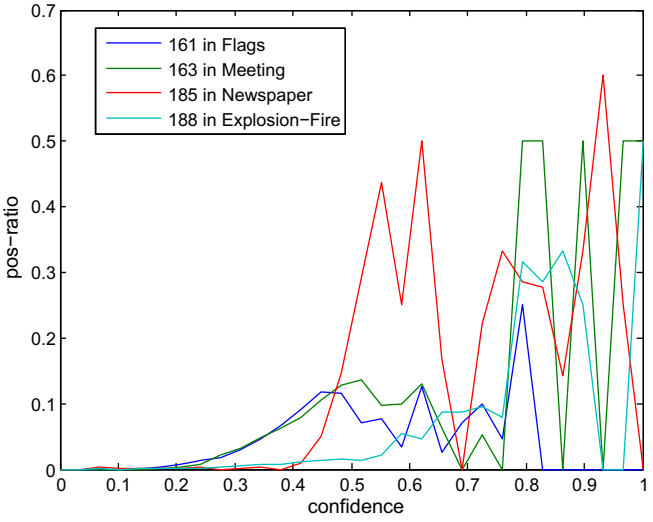


Fig. 6. The variation of the *rel-ratio* measure on some query ground truth in relevant concept threads.

Parameter N_c. Unlike leveraging on KL-divergence to automatically determine the number of related concepts [28], we just use *c-tf-idf* to filter $N_c + 2$ top concept candidates, and then choose the best N_c concepts with the largest p_b^+ and their respective threads. N_c could be selected on a per query basis. But we choose $N_c = 3$ for efficiency since no improvements can be observed after that point, as shown in Fig. 8. And we can see that SemanFeed is not sensitive to the number of concepts selected when $N_c > 3$.

Parameter N_v. Similarly, the number of query components are chosen as shown in Fig. 9. Again, the result is consistent over both TV05 and TV06 and we choose $N_c = 3$. Note that $N_v = 1$ corresponds

Given a rank list L and ground truth G:

(1) Finish if $N_u = 2000$ shots are browsed.

(2) Browse current rank list L in sequential order.

(3) When a relevant sample is found, its $N_b = 8$ temporal neighbors are browsed immediately. Turn to (2).

(4) Feedback action is triggered to generate a reranked list L' when either 20 relevant samples are newly collected, or when 200 results are browsed. Fuse L and L' as L. Turn to (1).

Fig. 4. The evaluation protocol.

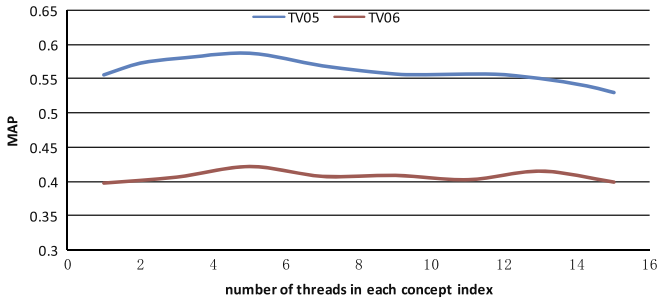


Fig. 7. Deciding the number of threads N_t in one concept.

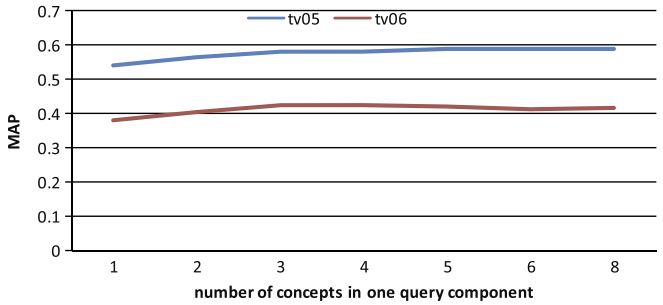


Fig. 8. Deciding the number of concepts N_c in one query component.

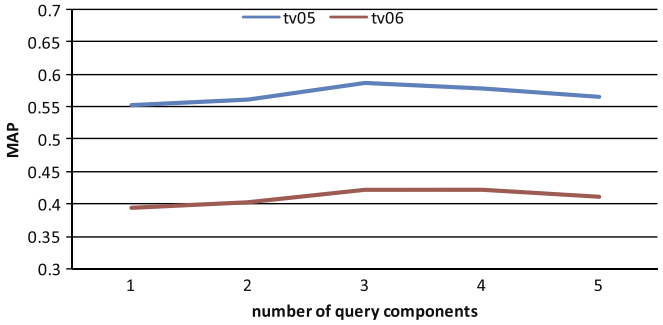


Fig. 9. Deciding the number of query components N_v given a query.

to concept thread without structured query formulation. The detailed evaluation of concept thread without structured query formulation is presented in Section 5.4, where for most query topics, structured formulation outperforms the single query component. So this structured formulation is more powerful in representing the query information need.

We list the structured concept thread formulation for three queries in Table 2. For "office setting", the second query component and the third one are supplements to the first component. For example the thread [0.6,0.8] of *Computer-TV-Screen* possibly contains a lot of computers and the *Report* and *Anchor* imply news studio which has high visual similarity with office setting. For "people reading newspaper", it is clear that the relevant threads of *Newspaper* are dominating. We also synthesize one topic "helicopter or ship/boat" by combining the ground truth of topic 0187 "helicopters in flight" and 0183 "water boats ships". The first two query components consist of *Lake*, *Waterway* and *Airplane-Takeoff*, which mainly captures the concept *ship/boat*. Then the structured formulation finds another component consisting of *Helicopter-Hovering*, *Smoke-Stack* and *Factory* with a clue about *sky*, thereby video clips about *Helicopter* possibly congregate in this component.

5.4. Comparative experiments

5.4.1. SemanFeed vs. c-tf-idf

For easy comparison, we decompose SemanFeed into two separate configurations:

1. *SF-Single*: SemanFeed with a single query component of concept-threads from $N_c = 3$ concepts with leading *c-tf-idf* scores.
2. *SF-Full*: SemanFeed with the structured query formulation, with $N_v = 3$ and $N_c = 3$.

The top-ranked automatic runs in the respective TRECVID benchmark are taken as the initial rank lists, respectively (TV05 NUS4, TV06 IBM_QCLASS). Here we stick to the concept311 index. Both conditions will be changed to test SemanFeed's sensitivity to initial rank list and concept index later. We establish two baselines for comparison. The first one is produced by browsing $N_t = 2000$ shots[3] following the protocol in Fig. 4, without executing the feedback step. The second one is feedback with the *c-tf-idf* approach. As previous experimental result shows [29], *c-tf-idf* works as good as the logistic regression model in IVR.

The per-query comparison results are shown in Fig. 10. MAP is also shown, together with an additional MP1 calculated as MAP excluding query 171 (195) on TV05 (TV06) since this nearly perfect query result cannot help to distinguish different approaches. It is evident that the two concept based approaches significantly improve the browsing only baseline both on TV05 (70%) and TV06 (55%), measured in MP1. The observation that SF-Full is 20% better than *c-tf-idf*, shows that SemanFeed explores the feedback data more fully. Note the large performance gap between *c-tf-idf* and SemanFeed in both query 185 and 189 can be explained by the former's inability to further identify relevant thread in the correct concept. For 185, the relevant examples are densely populated in [0.5,0.7] while the top of the index is filled with many close-up Newspapers. See Fig. 6 and Table 3 for further evidence. Similar phenomenon is observed for 189.

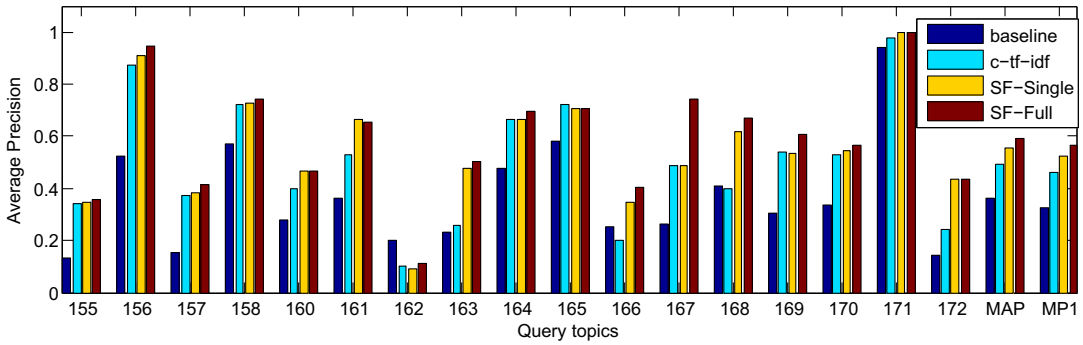
Comparison of SF-Full and SF-Single shows for queries with explicit or implicit OR logic, like 167, 168 and 169 in TV05, and 192, 193 in TV06, the performance increase is evident. For example, query 169 "tanks or other military vehicles" is clearly one with OR concept logic; query 193 "smokestacks, chimneys, or cooling towers with smoke or vapor coming out" is another one. Query 167 "airplane taking off" is one with implicit OR logic. Through carefully examining the concept index, we find that *Cigar-boat* selected in the second component has some water related appearance which resembles the airplane run into the sky. When no explicit OR logic is present in the query, the two-layer query inference network does not hurt the performance. Thus we consistently observe a 7% performance gain of SF-Full over SF-Single.

To understand why thread outperforms *c-tf-idf*, we pay close attention to the difference between the representations and perform one round feedback with both *c-tf-idf* and SemanFeed on the top 200 shots of automatic search result for each topic. For SemanFeed, 5 concepts are chosen as initial candidates and the top threads are generated. Results for a few queries with different overall relevant ratios are shown in Table 3. Although concepts selected by *c-tf-idf* are quite reasonable, we find that their weights do not change much. In contrast, the thread representation not only selects finer-granular concept threads, but also gives more diverse *rel-ratio* weights for the threads. The difference is evident in "people reading newspaper". For *c-tf-idf*, *Newspaper* and *Host* receive almost the same weight, while their *rel-ratio* weights are quite

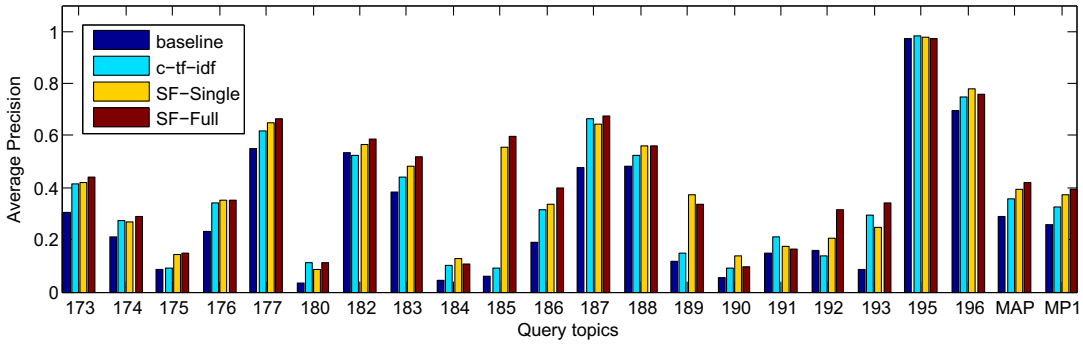
[3] A few topics have no more than 100 ground truth shots are browsed with more neighbor shots around the discovered relevant shots to get 2000 shots. However, this will not influence MAP too much because these queries usually have low APs.

Table 2
Three query components of "office setting", "people reading newspaper" and "helicopter or ship/boat". Only top 4 concept-threads are listed for each component here due to space limitations.

Topic	Query components			
Office setting, desks, tables, computers	Office [0.8,1.0] Office [0.8,1.0] Male-Reporter [0.8,1.0]	Office [0.6,0.8] Office [0.6,0.8] Male-Anchor [0.8,1.0]	Computers [0.6,0.8] Computer-Or-TV-Screens [0.6,0.8] Lawyer [0.6,0.8]	Computers [0.8,1.0] Reporters [0.6,0.8] Male-Anchor [0.6,0.8]
People reading newspaper	Newspapers [0.6,0.8] Newspapers [0.6,0.8] Newspapers [0.6,0.8]	Newspapers [0.8,1.0] Newspapers [0.8,1.0] Interview [0.8,1.0]	Newspapers [0.4,0.6] Newspapers [0.4,0.6] Newspapers [0.8,1.0]	Host [0.8,1.0] Host [0.8,1.0] Newspapers [0.4,0.6]
Helicopter or ship/boat	Lakes [0.8,1.0] Lakes [0.8,1.0] Helicopter-Hovering [0.8,1.0]	Waterways [0.8,1.0] Waterways [0.8,1.0] Helicopter-Hovering [0.6,0.8]	Airplane-Takeoff [0.8,1.0] Airplane-Takeoff [0.8,1.0] Smoke-Stack [0.8,1.0]	Waterways [0.6,0.8] Waterways [0.6,0.8] Factory [0.8,1.0]



(a) TV05 queries.



(b) TV06 queries.

Fig. 10. Per query comparison of different concept based feedback approaches.

Table 3
Sample queries and their respective representation by concepts/threads. Queries are listed in the descending order of relevant items found in the top 200 shots. For concept based representation, the top 3 selected concepts and their respective *c-tf-idf* score is shown. For thread based representation, the top 3 selected threads and their respective *rel-ratio* weights are shown.

Topics	Feedback statistics	Concept representation		Thread representation	
		Top 3 concepts	*c-tf-idf*	Top 3 threads	*rel-ratio*
Soccer goalposts	Relevant: 94 Irrelevant: 106	Soccer Sports Lawn	1.42 1.19 1.16	Soccer [0.8,1.0] Sports [0.8,1.0] Lawn [0.4,0.6]	0.94 0.93 0.9
Natural scene	Relevant: 39 Irrelevant: 161	Landscape Valleys Hill	0.58 0.57 0.56	Landscape [0.8,1.0] Hill [0.4,0.6] Mountain [0.4,0.6]	0.67 0.6 0.48
Group people, dressed in suits, seated, with flag	Relevant: 20 Irrelevant: 180	Protesters Bus Interview	0.4 0.3 0.29	Meeting [0.8,1.0] Interview [0.4,0.6] Meeting [0.4,0.6]	0.33 0.18 0.14
People reading newspaper	Relevant: 5 Irrelevant: 195	Newspapers Host News-Studio	0.61 0.57 0.5	Newspapers [0.4,0.6] Host [0.4,0.6] Guest [0.2,0.4]	0.5 0.07 0.05

different. For "group people, dressed in suits, seated, with flag", the implicitly relevant concept thread of *Interview* [0.4,0.6] is identified by SemanFeed while *c-tf-idf* can only search at the top of the rank list of *Interview*. This again shows that relevant results do not always cluster at the top of a albeit relevant concept rank list.

5.4.2. SemanFeed vs. ordinal reranking

Though SemanFeed clearly outperforms *c-tf-idf*, it is necessary to compare it with other reranking algorithms. Here we choose the recently proposed ordinal reranking [12] approach since it shows state-of-the-art performance in AVR and is superior to SVM with pseudo-bagging in the concept subspace [26]. The experiments are conducted on both concept311 and concept374. Following [12], we implement the linear ListNet algorithm [27]. The learning rate η is set as $\eta = 0.5$ since larger value leads to no convergence for some queries on concept311. We also enumerate the number of concepts selected by *c-tf-idf* from 5 to 100 concepts with increment 5 (including 75 concepts in [26]). We select the number of concepts with the best performance for fair comparison across approaches. The results are shown in Table 4.

From Table 4, we observe that SemanFeed consistently outperforms ordinal reranking on both datasets across these two concept lexicons. This is reasonable since ordinal reranking shares the same top ranking assumption with *c-tf-idf*. Actually it assigns different weights to the concept lists selected by *c-tf-idf*. In contrast, SemanFeed relaxes such assumption. To our surprise, ordinal reranking does not always outperform *c-tf-idf*. We conjecture that the absolute relevant/irrelevant judgements provided in IVR are harder to regress than normalized rank scores appeared in AVR [12] where ordinal reranking is initially applied.

5.4.3. SemanFeed across different concept detectors

Now we analyze the influence of different concept detectors on SemanFeed. This experiment is helpful for determining whether SemanFeed takes advantage of some detector specific characteristics or relies on more general cues to gain over *c-tf-idf*. If SemanFeed leverages on some particularity of the indexing mechanism of concept311, we should observe that its performance is similar to or even worse than that of *c-tf-idf* on concept374. Shown in Table 4, SemanFeed significantly outperforms *c-tf-idf* on both concept374 and concept 311, and with a large margin on 3 of 4 times. Though preliminary, this evidence is against that SemanFeed is concept-detector dependent. Rather, it is quite possible that SemanFeed utilizes some concept hierarchy and/or identifies query restricted concept thread through feedback information. However, given current limited query and associated groundtruth, we cannot draw definite conclusion beyond this. It is noticeable that the performance gain of SemanFeed over the baseline is smaller on concept374. However, we observe the same phenomenon on both *c-tf-idf* and ordinal reranking. We attribute this to the relative lower quality of concept374.

5.4.4. SemanFeed across different initial lists

Similarly, it is interesting to see how the initial rank list affects SemanFeed's performance. Thus besides the best lists, we incorporate two different initial rank lists from TV05/06 Tsinghua automatic runs which are only of better-than-average quality. Following the same feedback protocol, we obtain the results in Table 5. From Table 5, we have the following observations: (1) Over both initial rank lists, SemanFeed achieves significant performance gain over the simply browsing baseline and *c-tf-idf*. So SemanFeed is not very sensitive to initial result performance variations. (2) For the Tsinghua runs with only moderate initial performance, the performance gain is more impressive (99% on TV05 and 86% on TV06). Thus SemanFeed effectively utilizes the feedback information to reduce the performance gap between different initial rank lists.

5.5. Error tolerance and efficiency

5.5.1. Feedback dynamics

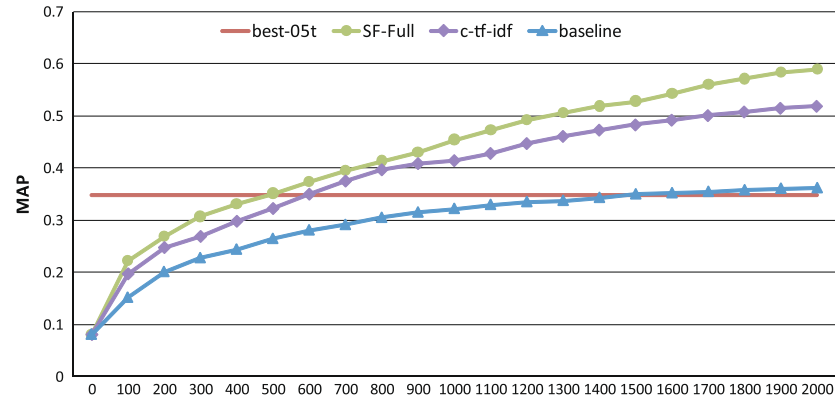
The feedback dynamics tell us more than the final MAP score. Three approaches of SF-Full, *c-tf-idf* and baseline are again compared. We fix the feedback round to 100 documents here for illustration purpose. The MAP for each round is shown in Fig. 11. We can see that SF-Full examines only 500 documents to achieve the full performance of the baseline in TV05, and some 900 documents in TV06. So inferring the latent user information need through concept feedback really helps alleviate the user burden. Consequently, this concept based feedback approach has great potential for IVR. The gap between SF-Full and *c-tf-idf* is evident, especially after a few rounds of feedback. The initial small gap in TV06 is because SemanFeed need more examples to estimate its additional parameters. The best submitted interactive runs in TRECVID are also plotted, but only for *reference*. Rather, we would like to emphasize that real interactive systems are hard to compare due to the complex factors involved such as user, interface and indexing mechanism. It is also important to point out that our simulated experiments are too optimistic in failing to recognize the human factors in retrieval and overlooking the errors users made. We simulate such labeling error subsequently. But rigorous user study is beyond the scope of this paper.

Table 4
Performance comparison of SemanFeed, *c-tf-idf* and ordinal reranking (denoted as ordinal in the table) on TV05/06 with two different sets of concept detectors. Percentage in the parentheses is the improvement over the respective baseline. Note MP1 is MAP excluding nearly perfect query 171 (195) on TV05 (TV06).

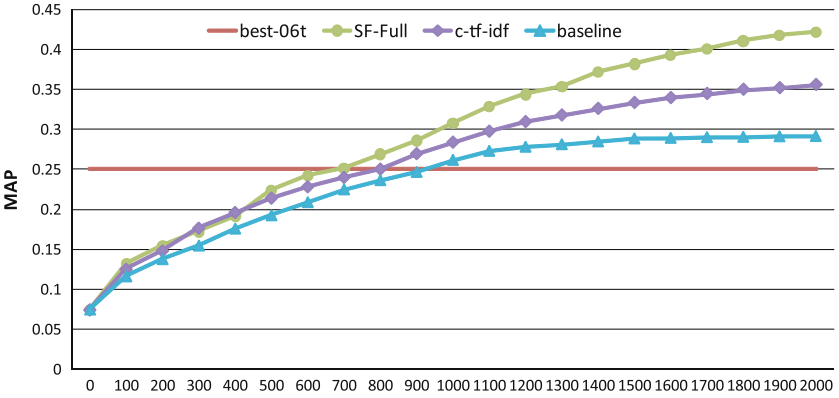
MP1	Baseline	Concept311 detector			Concept374 detector		
		c-tf-idf	Ordinal	SF-Full	*c-tf-idf*	Ordinal	SF-Full
TV05	0.326	0.461(41%)	0.474(45%)	0.564(73%)	0.434(33%)	0.449(38%)	0.470(44%)
TV06	0.256	0.323(27%)	0.319(25%)	0.393(54%)	0.283(11%)	0.335(31%)	0.350(37%)

Table 5
SemanFeed with two initial rank lists on TV05/06 of different performance. The good ranklists are only of better-than-average quality. Percentage in the parentheses is the improvement over the respective baseline. Note MP1 is MAP excluding nearly perfect query 171 (195) on TV05 (TV06).

MP1	Best rank list			Good rank list		
	Baseline	*c-tf-idf*	SF-Full	Baseline	*c-tf-idf*	SF-Full
TV05	0.326	0.461(41%)	0.564(73%)	0.253	0.434(73%)	0.505(99%)
TV06	0.256	0.323(27%)	0.393(54%)	0.170	0.273(61%)	0.316(86%)



(a) TV05 queries.



(b) TV06 queries.

Fig. 11. Feedback dynamics for different approaches, the *x*-axis is the number of examples examined.

5.5.2. Mislabeling tolerance

To inspect the capability of mislabeling tolerance, we allow some labeling noise. Analysis of previous real user log shows that users on average have 20% false alarm rate (irrelevant examples to be mistakenly labeled as relevant examples) and 3% miss rate (relevant examples to be mistakenly labeled as irrelevant examples). This is reasonable in the TRECVID scenario since users usually tend to submit all plausible result in the limited inspection time. Thus we perform simulative experiments on TV05 to allow for *r*% false alarm error in the feedback. For each *r*, the experiment is repeated 10 times. With those noisy feedbacks, we mine the concept-threads with the SF-Full model and retrieve the corpus with the threads. The performance is shown in Table 6. Considering that the falsely incorporated irrelevant examples will bring the MAP down to 0.190, the 0.175 MAP averaged on the noise contaminated 10 runs does not deteriorate the performance much beyond that point. So this approach is rather insensitive to relevant example labeling errors. The reason for this robustness comes from the average and smoothed counting operation in each query component and thread.

Table 6
Simulated IVR results with different mislabeling noise level.

Noise rate *r* (%)	0	10	15	20	25
MAP estimation	0.236	0.218	0.201	0.190	0.180
Average MAP	–	0.187	0.177	0.175	0.174

5.5.3. Time efficiency

Procedures of both mining structured concept-threads and reranking are efficient since only partial data are involved. It costs 100ms on average to return the fused search result from feedback. This processing time is tolerable for users in an interactive retrieval scenario. The experiments are all conducted on a standard laptop with 2.0 GHz Intel Core 2 Duo CPU and 2 GB memory. Note that the performances are obtained via an unoptimzied prototype system. Efficient inverted-list will proven to be a great speed-up for our application. This advantage makes our approach competitive for practical search engines where real-time execution is a must.

6. Conclusions

In this paper, we attempt to provide a preliminary investigation into the problem of alleviate the limited concept lexicon for expressing possible complex query needs. This problem is neither trivial nor ephemeral. To be concrete, we provide a new formulation for video query as structured combination of concept threads, contributing to the general *query-by-concept* paradigm. The proposed representation incorporates the previous concept based *c-tf-idf* formulation as a special case and extends the restricted AND concept combination logic to a two-level concept inference network. We apply this new formulation to interactive video retrieval on the TRECVID 2005 and 2006 data sets. As evidenced by simulative experiments, the proposed query formulation offers some 60% improvements over the simple browsing search baseline in nearly real time. It also has clear advantage over the *c-tf-idf* approach within 100 ms of unoptimized feedback execution and

achieves better result than the state-of-the-art online ordinal reranking approach. It not only alleviates user's workload significantly but also is robust to mislabeling errors.

Although the size of queries is too small to be definite in conclusion, our results suggest a promising new line of research to mine absent specific concepts from the readily available indexed general concepts. We are currently investigating the following directions: improving the multi-concept thread generating algorithm; incorporating other possible query operations in the structured formulation and even automatic concept combination structure determination; exploring advanced user interface with user study experiments for further evaluation; last but not least, mining absent specific concepts from the already indexed concepts.

References

[1] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.

[2] M. Naphade, J.R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, J. Curtis, Large-scale concept ontology for multimedia, IEEE Multimedia Magazine 13 (3) (2006) 86–91.

[3] C.G. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, M. Worring, Adding semantics to detectors for video retrieval, IEEE Transactions on Multimedia 9 (5) (2007) 975–986.

[4] S.-F. Chang, W. Hsu, W. Jiang, L. Kennedy, D. Xu, A. Yanagawa, E. Zavesky, Columbia university trecvid-2006 video search and high-level feature extraction, in: Proceedings of TRECVID workshop, 2007.

[5] X. Li, D. Wang, J. Li, B. Zhang, Video search in concept subspace: a text-like paradigm, in: Proceedings of CIVR, 2007, pp. 603–610.

[6] M. Campbell, A. Haubold, S. Ebadollahi, M.R. Naphade, A. Natsev, J.R. Smith, J. Tešić, L. Xie, IBM research trecvid-2006 video retrieval system, in: Proceedings of TRECVID, 2006.

[7] C.G. Snoek et al., The MediaMill trecvid-2006 semantic video search engine, in: Proceedings of TRECVID, 2006.

[8] A.G. Hauptmann, W.-H. Lin, R. Yan, J. Yang, M.-Y. Chen, Extreme video retrieval: joint maximization of human and computer performance, in: Proceedings of ACM Multimedia, pp. 385–394, 2006.

[9] C. Snoek, I. Everts, J. van Gemert, J. Geusebroek, B. Huurnink, D. Koelma, M. van Liempt, O. de Rooij, K. van de Sande, A. Smeulders, J. Uijlings, M. Worring, The MediaMill trecvid-2007 semantic video search engine, in: Proceedings of TRECVID workshop, 2007.

[10] M.G. Christel, Establishing the utility of non-text search for news video retrieval with real world users, in: Proceedings of ACM Multimedia, pp. 707–716, 2007.

[11] R. Agrawal, T. Imielinski, A.N. Swami, Mining association rules between sets of items in large databases, in: P. Buneman, S. Jajodia (Eds.), Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, pp. 207–216, 1993.

[12] Y.-H. Yang, W.H. Hsu, Video search reranking via online ordinal reranking, in: Proceedings of ICME, pp. 285–288, 2008.

[13] A. Amir, J.O. Argillander, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, J.R. Kender, C.-Y. Lin, M. Naphade, A.P. Natsev, J.R. Smith, J. Tesic, G. Wu, R. Yan, D. Zhang, IBM research trecvid-2004 video retrieval system, in: Proceedings of TRECVID workshop, 2005.

[14] C.G.M. Snoek, M. Worring, J.-M. Geusebroek, D.C. Koelma, F.J. Seinstra, A.W.M. Smeulders, The semantic pathfinder: using an authoring metaphor for generic multimedia indexing, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1678–1689.

[15] D. Wang, X. Liu, L. Luo, J. Li, B. Zhang, Video diver: generic video indexing with diverse features, in MIR'07: Proceedings of the International Workshop on Multimedia Information Retrieval, ACM, New York, NY, 2007, pp. 61–70.

[16] A. Haubold, A.P. Natsev, M.R. Naphade, Semantic multimedia retrieval using lexical query expansion and model-based reranking, in: Proceedings of ICME, 2006, pp. 1761–1764.

[17] S.-Y. Neo, J. Zhao, M.-Y. Kan, T.-S. Chua, Video retrieval using high level features: Exploiting query matching and confidence-based weighting, in: Proceedings of CIVR, 2006, pp. 143–152.

[18] C.G. Snoek, M. Worring, Are concept detector lexicons effective for video search? in: Proceedings of ICME, 2007, pp. 1966–1969.

[19] M.G. Christel, A.G. Hauptmann, The use and utility of highlevel semantic features in video retrieval, in: Proceedings of CIVR, 2005, pp. 134–144.

[20] A. Natsev, A. Haubold, J. Tesic, L. Xie, R. Yan, Semantic concept-based query expansion and re-ranking for multimedia retrieval: a comparative review and new approaches, in: Proceedings of ACM Multimedia, 2007, pp. 991–1000.

[21] C.G. Snoek, M. Worring, D.C. Koelma, A.W. Smeulders, A learned lexicon-driven paradigm for interactive video retrieval, IEEE Transactions on Multimedia 9 (2) (2007) 280–292.

[22] X.-Y. Wei, C.-W. Ngo, Ontology-enriched semantic space for video search, in: Proceedings of ACM Multimedia, 2007, pp. 981–990.

[23] A.P. Natsev, M.R. Naphade, J. Tesic, Learning the semantics of multimedia queries and concepts from a small number of examples, in: Proceedings of ACM Multimedia, 2005, pp. 598–607.

[24] W. Zheng, J. Li, Z. Si, F. Lin, B. Zhang, Using high-level semantic features in video retrieval, in: Proceedings of CIVR, 2006, pp. 370–379.

[25] W. Hsu, L. Kennedy, S.-F. Chang, Video search reranking via information bottleneck principle, in: Proceedings of ACM Multimedia, Santa Babara, CA, USA, 2006, pp. 35–44.

[26] L. Kennedy, S.-F. Chang, A reranking approach for context-based concept fusion in video indexing and retrieval, in: Proceedings of ACM CIVR, 2007, pp. 333–340.

[27] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, H. Li, Learning to rank: from pairwise approach to listwise approach, in: ICML'07: Proceedings of the 24th International Conference on Machine Learning, ACM, New York, NY, 2007, pp. 129–136.

[28] D. Wang, X. Li, J. Li, B. Zhang, The importance of query-concept-mapping for automatic video retrieval, in: Proceedings of ACM Multimedia, 2007, pp. 285–288.

[29] D. Wang, Z. Wang, X. Li, J. Li, B. Zhang, Mapping query to semantic concepts: Leveraging semantic indices for automatic and interactive video retrieval, in: Proceedings of ICSC, 2007.

[30] A.F. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and trecvid, in: Proceedings of International MIR workshop, 2006.

[31] J. Adcock, J. Pickens, M. Cooper, L. Anthony, F. Chen, P. Qvarfordt, Fxpal interactive search experiments for trecvid-2007, in: Proceedings of TRECVID workshop, 2008.

[32] O. de Rooij, C.G.M. Snoek, M. Worring, Query on demand video browsing, in: MULTIMEDIA'07: Proceedings of the 15th International Conference on Multimedia, ACM, New York, NY, 2007, pp. 811–814.

[33] Z. Wang, D. Wang, J. Li, B. Zhang, Learning structured concept-segments for interactive video retrieval, in: Proceedings of CIVR, 2008, pp. 57–66.

[34] R.A. Baeza-Yates, B.A. Ribeiro-Neto, Modern Information Retrieval, ACM Press/Addison-Wesley, New York/Reading, MA, 1999.

[35] A. Aizawa, An information-theoretic perspective of tf-idf measures, Information Processing and Management 39 (2003) 45–65.

[36] F. Nah, A study on tolerable waiting time: how long are Web users willing to wait?, Behaviour and Information Technology 23 (3) (2004) 153–163

[37] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

[38] K.M. Donald, A.F. Smeaton, A comparison of score, rank and probability-based fusion methods for video shot retrieval, in: Proceedings of CIVR, 2005, pp. 1330–1344.

[39] A. Yanagawa, S.-F. Chang, L. Kennedy, W. Hsu, Columbia University's baseline detectors for 374 LSCOM semantic visual concepts, Columbia University ADVENT Technical Report 222-2006-8, Technical Report, 2007.