Integrating Web Videos for Faceted Search based on Duplicates, Contexts and Rules

Zhuhua Liao[1,2,3], Jing Yang[1], Chuan Fu[1], Guoqing Zhang[1]

[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]Graduate School of the Chinese Academy of Sciences
[3] Key Laboratory of Knowledge Processing and Networked Manufacturing, College of Hunan, Xiangtan, China
{liaozhuhua, jingyang, chuanfu, gqzhang}@ict.ac.cn

Abstract: We propose a novel video integration architecture, INTERVIDEO, for faceted search on web-scale. First, we demonstrate that the traditional video integration techniques are no longer valid in face of such heterogeneity and scale. Then, we present three new integrating techniques to build a global relation schema for organizing web videos and aiding user to retrieve faceted results. Finally, we conduct an experimental study and demonstrate the ability of our system to automatically integrate videos and build a complete and concise high-level relation schema on large, heterogeneous web sites.

Keywords: video integration, local relation view, global relation schema, faceted search

1. Introduction

Since there has been exponential growth with the popularity of social media in Web 2.0, the video collection environments are leading to the need for flexible video retrieval systems which deal with adaptive, multi-faceted search [1]. Faceted search provides flexible access to information by one or more facets which represent dimensions of information (e.g., category, time and location). However, there are many challenges for such faceted search to web videos. First, the semantic knowledge of videos such as annotation is very sparse, where the problem of query answering with incomplete information is intractable. Second, there are lacks of integration approaches on multiple dimensions for relevant content which reside at different video sources to organize web videos and enrich video's knowledge.

Similar aspects of research can be found on faceted search [1,2], data integration [3,4] and video retrieval system [5,6]. However, the work of faceted search only focus on the faceted metadata and category-based interface design, but not the information organization with multi-facets, especially the web videos' organization; the traditional work of data integration were mostly based on deep-web sources and mapping or reformulating of heterogeneous data schemata, such as the Meta-Querier project [7] and the PayGo architecture [8]. Recently, many content sites can share structured

data to users and other web sites by initiatives like OpenID and OpenSocial, In [9], the authors propose the SocialScope to integrate data based on OpenID and OpenSocial. But in the all work, they do not consider video integration on heterogeneous and video collection with the features of sparse annotations and distributing discrete, nonintegrated videos on the Web. And the video retrieval system's work is only intended for matching by text, image, and concept, etc.

Video integration for efficient video search has two broad goals[4]: increasing the completeness and increasing the conciseness of relation view over video collections that is available to query and index to users and applications. An increase in completeness is achieved by adding more video sources (more videos, more attributes describing video) to the system and integrating sources that supply additional attributes to the relation. An increase in conciseness is achieved by removing redundant videos and links, and aggregating duplicates and merging common attributes into one.

The goal of our video integration system is to combine the annotations, contexts and various relations of relevant videos which residing at different sources, providing the user with a unified relation view, called *global relation schema*. User formulates queries over the global relation schema, and the system suitably queries the sources, providing complete, concise and faceted results to the user.

2. System Overview

This section describes the design and implementation of the INTERVIDEO system. INTERVIDEO is modeled as a client-server system, where the search clients interact with both web video sites and video integration server. The overall system architecture is presented in Figure 1. In the system, we first use information retrieving tools to retrieve video's annotations and relationships for building local relation view of videos. Then, we integrate various local relation views with new techniques to build global relation schema and refine it.

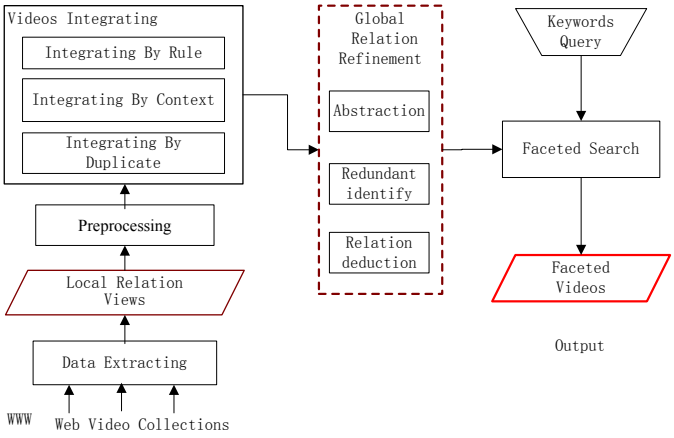

Figure 1. INTERVIDEO System Architecture

- **Local relation view retrieving**. In general, the intense semantic relationships of videos can be found in the published web pages. At present, many techniques have been proposed to mine and retrieve the relation links imbedded in web pages. In order to extracting local relation view from HTML codes we use information extracting tool.
- **Global relation schema building.** In order to building the global relation schema, we propose three classes of novel techniques. These are (1) duplicate-based integration technique which takes relationship of immediately duplicate to integrate videos and enrich video's annotations; (2) the context-based integration technique which leverages the contexts such as tagging to identify the relationships between videos; (3) the rule-based integration technique which uses rules that user specified to integrate videos. In the section 4, we will describe three techniques in detail.

3. Local relation view retrieving and duplicate detecting

Information extracting from HTML [10,11,12,13] is usually performed by software modules called *wrapper*. In most cases, a practicable wrapper should be able to identify the template, and hence extract the data fields from any new pages having the same template. In the system, we use the RoadRunner[10] and specify templates to extract local relation view from web pages on different web sites. The templates defined by HTML codes and compiled to a consistent specification in XHTML, a restrictive variant of HTML. The specification defines a set of interrelated entities: a video element links a set of duplicates videos and annotation; and a set of correlative videos with logic relation (e.g. sequential relation). The data extraction process is introduced in [10] in detail.

Among huge video collections with many near duplicate videos (X. Wu observe that on average there are 27% redundant videos[14]), efficient near duplicate detection is essential for effective search, retrieval and integration. We built a video duplicate detector to detect near duplicate [14] in video collections based on the work originally presented in [16]. This fingerprint-based method relies on robust hash functions, which take an input message (video frames in our case) and generate a compact output hash value, with the condition that similar input messages generate similar output values. All videos involved in the detection are converted into hash values, and detection is performed as a search problem in the hash space. The system uses the robust hash functions and search procedure which described in [16]. The precision-recall was verified approximately 0.8 [15].

4. Global relation schema building

In the paper, we consider the relation view of videos as a relational graph and the integration of relation views is equal to merge two or more graphs.

Definition 4.1 (Graph) A (relational) graph is a tuple G=(V, E,R,W) where V is a set of nodes, E is a set of edges, R is a set of relation of each edge, and W is a weight matrix of each relation. The relation set can be included similarity, time or space proximity, sequence etc.

We define an operator on graphs, Union, as follows:

Definition 4.2 Union($\cup$): Let G_i and G_j be two relational graphs that present the relation between videos. The $G_i \cup G_j = \{G \mid V=V_i \triangle V_j,\ E= E_i \triangle E_j,\ R= R_i \triangle R_j, W=W(R)\}$, where $\triangle$ is the operation of symmetric difference in logical algebra.

4.1 Duplicate-based Integration

Generally, the relation view of one duplicate represents a faceted semantic relation of the duplicate in bigger space. So the duplicate-based integration can help to build a global relation schema on video sources. In view of the neighbours of duplicate may be duplicate, we can not simple merge these local relation views. We consider eliminating the common nodes which represent the same video in these views. Algorithm **IntegrateByDuplicate** describes the steps in integrating local relation view G_i and G_j. At a high-level we first detect all duplicates between G_i and G_j and update the names of nodes that represent the duplicates for name consistency. Then we consider the pre-processing of relations and weights for relation consistency. The pre-processing of relations is included the relation transform, such as: the video A was created on "2010.4.9" and B was created on "2010.4.19", and there are exist the relation r_1="is same month" with w=1-(1/3) in G_i. But in G_j there are used the relation r_2="is same year", so for ensuring relation consistency in union view, r_1 can be transformed to r_2 with w=1-(10/365). Note that, we transform relation from old relation in combining views, but do not delete the old relation.

Algorithm 1 IntegrateByDuplicate(G_i,G_j: View of duplicate)

1: Dset=**DetectDuplicateNodes**(G_i, G_j);
2: **for** each duplicate **do**
  if there are duplicates between G_i and G_j **then** update the names of nodes that represent the duplicates to the same but different with other non-duplicates' nodes;
  end for
3: Preprocessing:
  If $r_i(R_i) \subset r_j(R_j)$ or $r_j(R_j) \subset r_i(R_i)$ **then** do relation transform; **End if**
  If $r_i(R_i)$ is the same as $r_j(R_j)$ and $w_i(r_i) \neq w_j(r_j)$ **then**
    $w_{r_i} = w_{r_j} = (w_{r_i} + w_{r_j})/2$;
    End if
4: $G = G_i \cup G_j$;
5: **return** G;

Note that the algorithm **IntegrateByDuplicate** is the main idea that integrating two local relation views by near-duplicate. In the whole video collections, if there are multiple local relation views and duplicates between them, the algorithm **IntegrateByDuplicate** will be called repeatedly until there are not duplicates in all local relation views.

4.2 Context-based Integration

Although no near duplicates in some local relation views, we observed that some of

videos in different views will similar in semantics if their annotations such as tagging, description are very similar. In the paper, we take the annotation and comments of a video as context of the video. On account of integrating these videos and their relevant videos can help to retrieve bigger relation view, the integration based on context is useful technique for our system. Algorithm **IntegrateByContext** summarizes that how we integrate the graph G_i and G_j if we find there are relations with highly weight W_{ij} between nodes v_i ($v_i \in G_i$) and v_j ($v_j \in G_j$).

Firstly, in algorithm 2, we integrate the duplicates by using the algorithm **IntegrateByDuplicate** for merging the same videos. Then deducing the relation type of videos (such as similarity, sequence) between G_i and G_j in which these videos have same attributes or keywords in the context. In the step of relation establishment, we establish the directional relation for sequential relation, and the similar relation with the similarity computing technique [16] to compute the similarity of video's annotation as the weight. Note that for determining what similarity of context between videos is considered to be integrated, we use a threshold of the similarity σ which can be set by user or system.

Algorithm 2 IntegrateByContext(G_i,G_j: Graph)

1: Firstly, integrating by duplicates:
  G= IntegrateByDuplicate(G_i,G_j);
2: Finding same attributes or keywords in the context of V_i and V_j
 Deducing the relation type between V_i and V_j
  If there are existing relation and not edge between V_i and V_j **then**
    Generate a edge between V_i and V_j
  End if
 If there are sequential relation **then** Establish the directional relation r_{ij}; **end if**
  Else if there are similar relation **then**
    Computing similarity of the values of attribute both in V_i and V_j on same attribute
    If the similarity great than the value of a threshold that user set **then**
      Establish the relation r_{ij} and assign the similarity to $w(r_{ij})$
    End if
  End if
3: **return** G;

4.3 Rules-based Integration

The techniques introduced above can automatically integrate correlative videos by duplicate or context. There is one type of video integration which can not be integrated with obviously correlative relationship, but can be integrated with logic rules, such as constituent, time or space distance, etc. In general, there are mainly two classes of rules: (1) Numerical Rules that integrate a set of videos by a numerical bound; (2) Set Rules that integrate a set of videos by enumerative tags.

Definition 4.3 Numerical Rule. Let S be a set of videos, *A* be a set of common attributes of *S*, i.e. $A=\{a_1,a_2,a_3,\ldots\}$, the simple rule $R_s=\{V|\ S.a_i \otimes E\}$, where $\otimes$ is one of <, =, >, E is a expression which limit the bound, and V is the videos that satisfied the rules.

As an example, Let *A* be a set of common attributes of a video collection S, such as load time (*lt*), length, and so on. The function $R_s=\{V|\ S.lt>\text{DATE}\}$ integrates the set

of all video that their load time late than the DATE.

Definition 4.4 Set Rule. Let S be a set of videos, *A* be a set of common attributes of S, i.e. $A=\{a_1,a_2,a_3,\ldots\}$, the set rule $R_s=\{V|\ S.a_i \oplus T\}$, where $\oplus$ is similarity operator, and T is a set of enumerative phrases.

It is easy to see that sometime there are not a video's content covered a subject, but a set of videos, so user can specify a set of sub subject names to query.

The algorithm **IntegrateByRules** describes how we integrate videos based on some rules that user given. At first, we use the traditional match algorithms e.g. Vector Space Match [17] to select the videos V that satisfied numerical rules and set rules, then generate edges for V and merge the graphs of these videos.

Algorithm 3 IntegrateByRules(R_s: specific rules; S: video set)

1: $\forall r_i \in R_s$; G ={};
2: **if** r_i is Numerical Rule **then**
3: V = $S.a_i \otimes E$;
4: **else if** r_i is Set Rule **then**
5: V = $S.a_i \oplus T$;
6: **end if**
7: **for** $v_i, v_j \in V$ **do**
8: G_i =the graph of v_i; G_j =the graph of v_j;
9: e_{ij}=edge between v_i and v_j;
10: G=G□G_i□G_j;
11:**end for**
12:**return** G;

5. Global relation refinement

Using the algorithms of section 4, we can build the global relation schema on video collections, but the global relation schema is complex, and there are redundant and conflicted relations which will impede the faceted search seriously. In the section, we will consider the abstraction of nodes, redundant relation rectification and relation deduction for refining the global relation of video collections.

(1)**Nodes abstraction**. The name of some relations has implicitly declared that the nodes belong to one category or have same feature, such as "is same (time, color, etc)", "is belong to (common command in computer networks, electronic commerce course, etc)". So we can build an abstract node to link these nodes and use the same feature and category name as its name which shows in figure 2(a). To some abstract nodes if they be included a wider category or have common features, we can build an abstract nodes on these abstract nodes which show in figure 2(b).

(2)**Redundant identify**. Although in the integrating process, we try to integrate all duplicate videos or videos with same tags. But it is hard to keep the global relation schema with no redundancy. There are may be some relations with different their name but they are the same relation in real, so in the global relation schema we need to identify the redundant relations by the semantic analysis, such that synonyms, alias, etc.

(3)**Relation deduction**. Some relations between a set of videos have the transitive

or symmetrical characteristics. For videos a, b, c, that is:

if $a \xrightarrow{r} b$ then $b \xrightarrow{r} a$ (symmetry);

if $a \xrightarrow{r} b$ and $b \xrightarrow{r} c$ then $a \xrightarrow{r} c$ (transitivity).

So we can deduce:

r is symmetrical relation and $a \xrightarrow{r} b \Rightarrow b \xrightarrow{r} a$;

r is transitive relation and $a \xrightarrow{r} b$, $b \xrightarrow{r} c \Rightarrow a \xrightarrow{r} c$.

By the relation deduction for transitive or symmetrical relations, we can complement the relations that implied in videos.

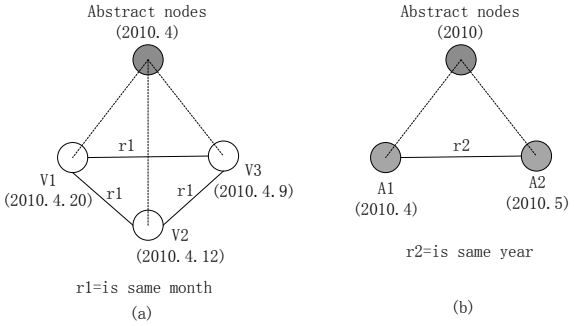

Figure 2. Nodes abstraction

In short, through the nodes abstraction, redundant rectify and relation deduction, we will get a compact, clear and hierarchical global relation schema which will make the faceted search became effectively, quickly and completely.

6. Experiments

We conducted an experimental study the performance of the system. In the experiment, we mainly consider two integrating techniques: duplicate-based integration and context-based integration. The goal of the study was to understand the effect of our technique to integrate videos on heterogeneous video collections and the contributions of the various constituents in the system.

6.1 Experimental Setup

All the video set in our experiments are crawled by searching on the Google Video and Yahoo!Video. We select 80 popular keywords about “Computer Networks” topic as the queries (in Google Video we using Chinese keywords). For each query, we get 100 top-ranked videos and their corresponding web pages. We refer to the dataset from Google Video and Yahoo!Video as GV and YV respectively. We use the RoadRunner to extract local relation view from web pages. Then we use the two integrating techniques to build the global relation schema.

To measure the effectiveness of our techniques for automatic video integration, we

perform video integrating to estimate the conciseness, completeness, integration gain respectively and compared the video systems of Google, Yahoo in the following experiments by randomly select 10 keywords.

6.2 Effect of extensional conciseness

The conciseness measures the uniqueness of videos representations and boosting the video tagging, as well as the capability of eliminating copy, in video collections. Referred to [4], we define the extensional conciseness (EC) is the number of unique videos in a collection in relation to the overall number of video representations in the collection.

$$EC = \frac{\| \text{unique videos in video collection} \|}{\| \text{all videos in video collection} \|} = \frac{a}{a+b} \quad (1)$$

The example in the figure 3 shows the EC on the INTERVIDEO based on the experimental dataset of 10 keywords queries from GV and YV respectively. We observed that we can get EC=83.5% by our system. And further, we use the method of Nodes Abstraction (NA) to integrate all segments of videos, for example, using the "common command in computer networks" to representing the "part 1 of common command in computer networks" and the "part 2 of common command in computer networks", we can reduce more the EC, which is displayed as NA on GV and YV respectively in figure 3.

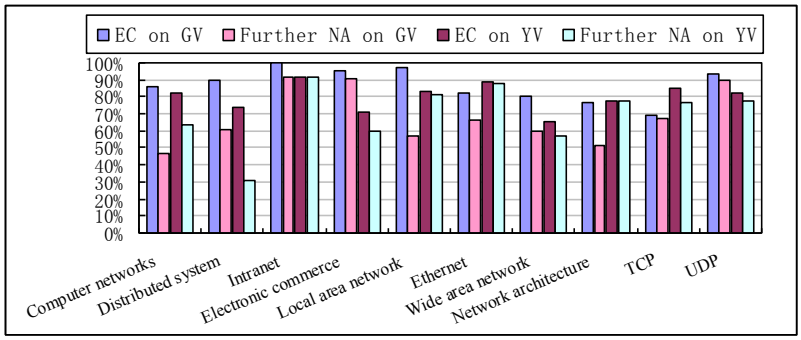

Figure 3. Measuring the EC on the INTERVIDEO

6.3 Effect of extensional completeness

The extensional completeness (EP) is the number of unique video representations in a dataset in relation to the overall number of unique videos in the Web, such as, in all the sources of an integrated system, referred to [4]. It measures the percentage of videos, which in the Web, covered by that dataset. We assume that we are able to identify same videos on the Web, for example, by an identifier created during duplicate detection.

$$EP = \frac{\| \text{unique videos in video collection} \|}{\| \text{all unique videos in the Web} \|} = \frac{a}{a+c} \quad (2)$$

In order to evaluate the EP, we not only use the GV and YV but also retrieving the videos that queried by Google Web. If we consider only the "intense relevant videos",

which is meaning the videos belong to the semantic space of the keywords, we observed that the EP equal to or slightly larger than 1. Because these dataset most from popular video website (e.g. www.youtube.com), in which the relevant videos in same web page is queried by same keywords in most case. But if we take the videos that queried by Google Web as experimental dataset, we can get high EP which in general the value great more than 1, and in most case the value can get to 5~8. We observed that the relevant videos with the videos we queried same in the web page is predefined and with same topic in these case. Note that the results that all returned by system together with topics but no discrete and disorder.

6.4 Integration gains

The integration gain (IG) is measuring average size of connected graph compared before and after video integrating. It evaluates the ability of interlinking with various semantic dimensions to a system.

$$IG = \frac{\text{average size of connected graph before integrating}}{\text{average size of connected graph after integrating}} \quad (3)$$

Generally, the videos queried by search engine are discrete and incomplete, and relevant videos are not linked. In our system, we can integrate the discrete videos with sorted and interlink to groups. The figure 4 shows the IG from our system with GV dataset, which has not been processed by global relation refinement. The results indicate that the results will be more semantic integration ability and comprehensive by our integrating techniques.

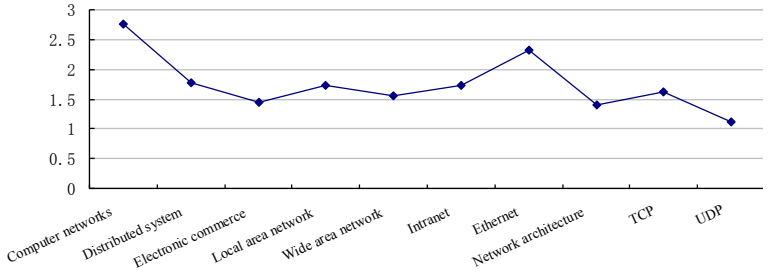

Figure 4. The Integration Gains (IG) of 10 queries in GV

7. Conclusions

In this paper, we have proposed a novel video integration framework for faceted search on the Web. More specifically, in what is a novel hybrid approach, we have used near duplicates, correlative contexts and specified rules to build global relation schema over heterogeneous video collections. The global relation schema which involves various relations and rich knowledge of videos enables faceted search. Our experiments show that the relevant videos fusion can largely improve concisely and completely structure and organization of content; our preliminary evaluation indicates an information gain and efficiency for videos searching. In the future, we plan to

resolve the integration conflict, which include the schematic conflict, and data conflict, etc. We also plan to automatically generate faceted metadata based on the global relation schema to boost the query refinement or results presentation.

8. Acknowledgement

We are grateful to the National High-Tech Research and Development Plan of China under Grant No. 2008AA01Z203 for funding our research.

References

1. K. P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In Proc. of the SIGCHI conference on Human factors in computing systems, 2003.
2. J. Teevan, S. T. Dumais, Z. Gutt. Challenges for Supporting Faceted Search in Large, Heterogeneous Corpora like the Web. Proceedings of HCIR, 2008.
3. G. Barish, Y. shin Chen, D. Dipasquo, C. A. Knoblock, et al. Theaterloc: Using information integration technology to rapidly build virtual applications. In ICDE, 2000.
4. J. Bleiholder and F. Naumann. Data fusion, ACM Computing Surveys, Vol. 41, No. 1, Dec, 2008.
5. J. Cao, Y. D. Zhang, et al, VideoMap: An Interactive Video Retrieval System of MCG-ICT-CAS, CIVR'09, July, 2009.
6. M. G. Christel, R. Yan. Merging Storyboard Strategies and Automatic Retrieval for Improving Interactive Video Search, CIVR'07, July, 2007.
7. K. Chang, B. He and Z. Zhang. Toward large scale integration: Building a MetaQuerier over database on the web. In CIDR, 2005.
8. J. Madhavan, S.R. Jeffery, S. Cohen, etc. Web-scale data integration: you can only afford to pay as you go. In CIDR, 2007.
9. S. Amer-Yahia,L. Lakshmanan and C. Yu. SocialScope: Enabling Information Discovery on Social Content Sites [C]. In CIDR, 2009.
10. V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards automatic data extraction from large web wites. In VLDB, 2001.
11. A. Arasu and H. G. Molina. Extracting structured data from Web pages. In SIGMOD, 2003.
12. Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In WWW, 2005.
13. M. Hung, Y. Zou. Recovering workflows from multi tiered e-commerce systems. 15th IEEE International Conference on Program Comprehension (ICPC'07), 2007.
14. X. Wu, A. G. hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search, In ACM Multimedia, MM'07, 2007.
15. S. Siersdorfer, J. S. Pedro, M. Sanderson. Automatic video tagging using content redundancy. SIGIR'09, July 19–23, 2009.
16. J. S. Pedro and S. Dominguez. Network-aware identification of video clip fragments. In CIVR '07, pages 317–324, New York, USA, 2007. ACM Press.
17. R. Abbasi, S. Staab. RichVSM: enRiched vector space models for folksonomies. Proceedings of the 20th ACM conference on Hypertext and hypermedia, 2009.