Web Video Topic Discovery and Tracking via Bipartite Graph Reinforcement Model

Lu Liu
Department of Computer Science and Technology
Tsinghua University
Beijing, 100084 China
lu-liu@mails.tsinghua.edu.cn

Lifeng Sun
Department of Computer Science and Technology
Tsinghua University
Beijing, 100084 China
sunlf@mail.tsinghua.edu.cn

Yong Rui
Microsoft China R&D Group
49 Zhichun Road
Beijing, 100080 China
yongrui@microsoft.com

Yao Shi
Department of Computer Science and Technology
Tsinghua University
Beijing, 100084 China
shiyao00@mails.tsinghua.edu.cn

Shiqang Yang
Department of Computer Science and Technology
Tsinghua University
Beijing, 100084 China
yangshq@mail.tsinghua.edu.cn

ABSTRACT
Automatic topic discovery and tracking on web-shared videos can greatly benefit both web service providers and end users. Most of current solutions of topic detection and tracking were done on news and cannot be directly applied on web videos, because the semantic information of web videos is much less than that of news videos. In this paper, we propose a bipartite graph model to address this issue. The bipartite graph represents the correlation between web videos and their keywords, and automatic topic discovery is achieved through two steps – coarse topic filtering and fine topic re-ranking. First, a weight-updating co-clustering algorithm is employed to filter out topic candidates at a coarse level. Then the videos on each topic are re-ranked by analyzing the link structures of the corresponding bipartite graph. After the topics are discovered, the interesting ones can also be tracked over a period of time using the same bipartite graph model. The key is to propagate the relevant scores and keywords from the videos of interests to other relevant ones through the bipartite graph links. Experimental results on real web videos from YouKu, a YouTube counterpart in China, demonstrate the effectiveness of the proposed methods. We report very promising results.

Categories and Subject Descriptors
H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based services*

General Terms: Algorithms, Experimentation

Keywords: web videos, topic discovery, topic tracking, bipartite graph model, reinforcement, co-clustering

1. INTRODUCTION
Thanks to the fast advancement of multimedia technology and increasing availability of network bandwidth, the volume of web videos is growing explosively in the past several years. Further fueled by the social media in Web 2.0, online video-distributing websites are attracting more and more web users. For example, YouTube [7], one of the most popular video-distributing websites in the world, announced that each day its users upload about 65,000 new videos and view more than 100 million videos [8]. In China, many YouTube-style video-distributing websites are emerging in the past two years, e.g. TuDou [10], YouKu [9], 6Room [11] etc. Market analysis data indicates that the number of video-distributing websites in China has grown to 150 in just one year (2006) [24]. Quite a few of them are growing very fast. Let's take YouKu as an example. 1). In terms of the website traffic, YouKu grows from nobody to No. 110 in the world and No. 11 in China in about a year [13]. 2). There are about 0.4% of internet users in the world that have visited YouKu. 3). On average, about 12 unique pages of YouKu are viewed per user per day.

The large number of video-distributing websites results in a fast expanding web video pool. This, unfortunately, leads to several difficulties: 1) it is difficult for end users to find what they are interested in; 2) it is difficult for websites administrators to organize the massive databases; and 3) it is difficult for advertisers to select which video to use.

If "hot" topics can be discovered and tracked automatically, it will benefit all the three types of people above. For example, users can easily find videos on "hot" topics in nearby days, and review the topics' history and development. Website administrators can automatically organize the videos and provide "value-added" services such as recommending relevant videos. Finally, topic tracking is useful for advertisers to analyze the relationship between the trace of topic development and the user behavior. Base on the analysis, the advertisers can decide on the most suitable type of advertisements and display them at the right time.

Topic detection and tracking is not a new field. But the domain has mostly been on news sources. Since NIST first proposed the problem of Topic Detection and Tracking (TDT) in the 1990s [17], a lot of work has been done. As the first effort, researchers proposed many approaches to detect and track topics on news documents [18] [19] [21] [22] [25]. Most of these methods are

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

based on either vector space models [25] [22] or statistical models [18] [19] [21]. For example, Yang et al. [25] represented news documents as vectors of words weighted by term-frequency inverse-document-frequency (TF-IDF) and used cosine angle to measure their similarity. Larkey et al. [19] estimated news documents' relevance language model and measured their similarity based on the asymmetric clarity-adjusted divergence. In recent years, researchers turn to multi-channel news videos for topic tracking [4] [14] [15] [26]. Most researchers used automatic speech recognition (ASR) technology to obtain the textual information from videos' content as the basic feature. Then they fused other visual features to assist the text-based methods. For example, Kender et al. [15] incorporated both the visual concepts and temporal properties to measure the video similarity. Hsu et al. [4] fused low-level features and visual near-duplicates in addition to the visual concepts. Zhai et al. [26] used facial region information and key-frames' global affine matching results as visual features. The experiments show that for news videos the textual features carry most useful information and the visual features add only marginal improvement.

The previous text-based approaches are all one-direction models, which means that they treat textual information as features and use them to represent documents/videos. This one-way model works well on the text-rich domain but may fail on the text-limited area for it results in very sparse representations. News documents obviously contain a lot of words. TV news programs can also obtain enough textual information from the speech content by ASR, because the videos are professionally edited and are in good quality. However, the textual information in web videos is very limited. As most of the web videos are amateur-made with varying quality, the technology of ASR cannot be readily used to extract useful semantic information from the speech content. The only direct textual information is videos' title and a few tags (we call them keywords) annotated by their users. Even worse, the limited textual information can be noisy and unreliable for two reasons:

- First, some users will annotate some "hot" but irrelative tags to attract more viewers. For example, a movie named "黄金甲 (Golden Armor)" was very popular in China last year, and we found that some videos which were irrelevant to "黄金甲 (Golden Armor)" were also annotated with "黄金甲(Golden Armor)".
- Second, due to people's different background and the lingual ambiguity, similar content can be annotated differently, and different content may be given the same tags. For example, the videos on the topic "Super girl" - a very popular annual national singing contest in China, can be annotated "超女 (super-girl)" or "超级(super)" "女声(voice)" or "超级 (super)" "女生(girl)" . On the other hand, "超级(super)" may be used to annotate other videos which have nothing to do with "Super girl".

To summarize, while topic detection and tracking is not a new field, its domain has been on news document/videos in the past. The web videos bring in a lot of new challenges including less semantic content and more noisy data, as analyzed above.

In this paper, we are motivated to investigate the problem of topic discovery and tracking on web-shared videos from YouKu [9], one of the most popular video-distributing websites in China. In order to overcome the problem of limited and noisy textual information, we propose a bipartite graph model to utilize the bi-direction correlation between the videos and the keywords. The basic idea is that the videos not only can be represented by keywords, but also can be used as features to propagate the textual information. The correlation structures between videos and keywords can also be analyzed to reduce the textual noise. Finally, the refined textual information in turn benefits and improves the performance of topic discovery and tracking. This framework forms an iterative feedback cycle.

The proposed framework is shown in Fig. 1. The topic discovery is achieved by two steps – coarse topic filtering and fine topic re-ranking. First, the information-theoretic co-clustering [3] is employed to filter web video topics at a coarse level. This is an unsupervised algorithm which utilizes the co-occurrence table of the two modules in the bipartite graph and obtains the clusters of videos and keywords simultaneously. In order to reduce noisy keywords' influence, we propose a weight updating strategy, which assigns each keyword a weight to reflect its impact on the co-occurrence table and updates the weights iteratively based on the videos' clusters information. Then, the videos on the discovered topics are re-ranked by analyzing the bipartite graph's link structures, which can be implemented as an iterative reinforcement process. The re-ranking step can be treated as a fine topic filtering step, because based on the re-ranking results, websites organizers can recommend the top N videos to customers and remove the videos with the least relevance.

After the topics are discovered, the interesting ones can also be tracked over a period of time using the same bipartite graph model. The basic idea is to propagate the relevant scores from pre-defined videos and keywords to other relevant ones through the bipartite graph's links, which can be also achieved by an iterative reinforcement process. After convergence, the relevant videos will be ranked higher than irrelevant ones.

The remainder of the paper is organized as follows: Section 2 briefly discusses the bipartite graph model. The two steps of topic discovery - coarse topic filtering and fine topic re-ranking are described in Sections 3 and 4 respectively. The topic tracking algorithm is discussed in Section 5. We report experimental results in Section 6 and give concluding remarks in Section 7.

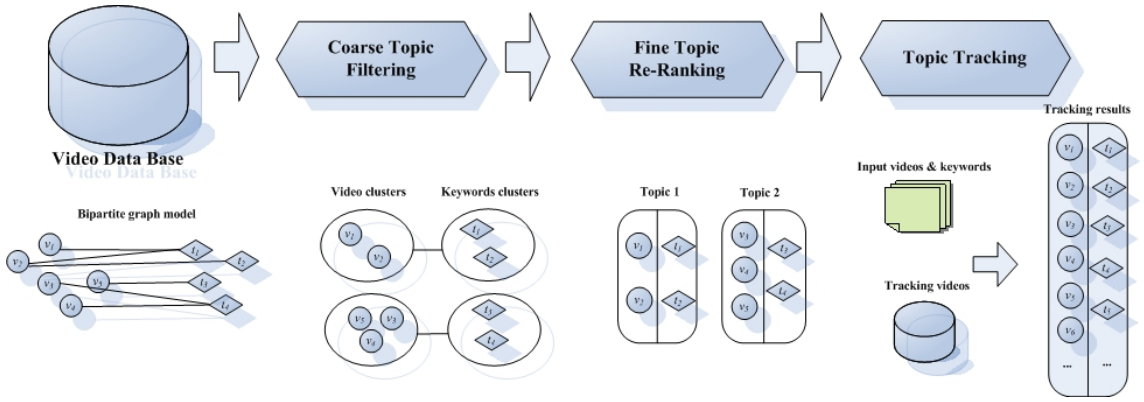

Figure 1: The framework of topic discovery and tracking by bipartite graph model

2. BIPARTITE GRAPH MODEL

As the keywords of web videos are limited and noisy, it's essential to use the bi-direction correlation between videos and keywords to enrich the textual information: re-assigning the keywords' weights, propagating the semantic and removing the noise. Inversely, the enriched textual information will make the videos' representation more effective and improve the performance. For example, Video 1, 2, 3, 4 have similar content and their keywords are shown in Table 1. If keywords are represented by using videos as shown in Table 2, it's easy to get that the keywords "Rockets; Yao Ming; NBA; basketball" have close relationship, so although Video 4 doesn't have keyword "basketball", it can be also found when we track the topic.

In this paper, we propose a bipartite graph model to represent the correlation between videos and keywords as shown in Fig. 2. There are two sets of nodes in the graph, which represent videos and keywords respectively. The links stand for the co-occur times between videos and keywords. The bipartite graph can be represented as a co-occurrence table C, of which the element $C(v, t)$ is the times keyword t occurs in video v .

Table 1: The keywords of 4 videos on the topic "basketball"[1]

Video 1	Rockets; Yao Ming; Star; humor series; NBA
Video 2	dunk; NBA; basketball
Video 3	NBA; newsreel; basketball; Yao Ming
Video 4	Yao Ming; Jazz; star; Rockets; NBA

Table 2: The keywords' representation

Rocket	Video 1, Video 4
Yao Ming	Video 1, Video 3, Video 4
NBA	Video 1, Video 2, Video 3, Video 4
Basketball	Video 2, Video 3

2.1 Keywords Selection

Web videos' keywords are obtained from the titles and tags. For Chinese's characteristic, the titles and tags are parsed by a natural language processing (NLP) [12] tool first to unify the form and then the participle results compose the keywords set.

As many keywords are meaningless and noisy, they need to be selected first. The keywords are filtered through two processes – word type filtering and mutual information filtering [4]. The former process is to filter out the stop words and other words with ambiguous word types such as adjective, adverb, etc. The latter process measures the mutual information between videos and keywords and filters out the keywords with small information value. The measurement is as Equ. (1).

$$IE(t) = p(t)\sum_{v} p(v \mid t) \log \frac{p(v \mid t)}{P(v)} \quad (1)$$

where t is the keyword, v is the video in the dataset. Actually, it has the same effect as removing the high-frequency and low-frequency keywords.

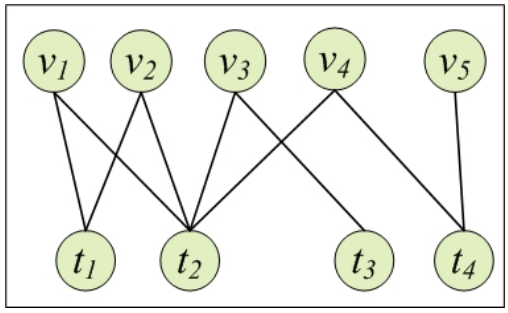

Figure 2: Bipartite graph model

3. COARSE TOPIC FILTERING

3.1 Information-Theoretic Co-Clustering

Co-Clustering is a kind of unsupervised algorithms which utilize the duality between two modules. Some co-clustering approaches have been proposed in literatures. For example, the work [2] is based on the spectral graph partition while the work [3] utilizes information theory. In our approach, information-theoretic co-clustering is adopted because it has less restriction and meets our needs.

Suppose there are m videos represented by a discrete random variable V whose value v is taken in the set $\{v_1, v_2, \ldots, v_m\}$ and there are n keywords represented by the other discrete random variable T whose value t is taken in the set $\{t_1, t_2, \ldots, t_n\}$. Let $p(V, T)$ denote the joint probability distribution between V and T. As V and T are both discrete, $p(V, T)$ is in nature the $m*n$ matrix, whose element is represented as $p(v, t)$. In our case, such a matrix can be obtained from the co-occurrence table C easily.

Suppose the videos belong to k topics and the corresponding keywords belong to l clusters. Thus V and T need to be grouped into k and l clusters, denoting as $\{vc_1, vc_2, \ldots, vc_k\}$ $\{tc_1, tc_2, \ldots, tc_l\}$

[1] We translated the Chinese tags into English.

respectively. These clusters could also be regarded as being generated by two discrete random variables *VC* and *TC*.

From the view of information theory, a fundamental quantity that measures the amount of information shared between *V* and *T* is the mutual information *IM*(*V*; *T*).

$$IM(V;T) = \sum_{v}\sum_{t} p(v,t)\log\frac{p(v,t)}{p(v)P(t)} \quad (2)$$

Work [3] indicates that an optimal co-clustering should minimize the loss of mutual information after clustering which satisfy Equ. (3)

$$\arg\min_{VC,TC}\{IM(V;T) - IM(VC;TC)\} \quad (3)$$

Towards the goal, work [3] proposes a four-step iterative co-clustering algorithm which is proved to decrease the loss of mutual information monotonically and guarantee to converge to a local minimum. Thus it can be employed to cluster the videos and keywords simultaneously.

3.2 Weight Updating Strategy

In order to distinguish keywords' different effect values, our scheme considers their weights on the joint probability. Thus, the probability is calculated by Equ. (4).

$$p(v,t) = \frac{w(t)C(v,t)}{\sum_{v,t} w(t)C(v,t)} \quad (4)$$

where $w(t)$ is the weight of t.

However, it is not easy to determine the weights before co-clustering. Furthermore, the weights actually depend on videos from which we want to discover topics. Thus we propose a method to update the weights iteratively based on the correlation between videos and keywords.

The weights are updated based on the results of video clusters. The main idea is that if a keyword dominates in a video cluster, it must be an important word for a topic, so it should be assigned a larger weight; however, if a keyword dominates in many video clusters, it's a common word and its weight should be decreased. So the weight updating idea works like TF-IDF measurement in information retrieval, but it takes the videos clusters as documents instead.

Suppose $f(t, vc_i)$ represents the frequency of keyword t in video cluster vc_i.

$$f(t,vc_i) = \frac{N(t,vc_i)}{|vc_i|} \quad where \quad N(t,vc_i) = \sum_{v\in vc_i} I(v,t) \quad (5)$$

where $I(v, t)$ equals to 1 if t occurs in the video v, 0 otherwise. $|vc_i|$ is the sum of all the keywords' occurrence in vc_i. Let $tf(t)$ denotes the maximal frequency of t, and $df(t)$ denotes the number of video clusters t appears. Then the keywords' weights are updated as in Fig. 3. The threshold thd_1 and thd_2 is set for reducing noise. The weight-updating co-clustering is shown in Fig. 4. In practice, a few rounds of updates yield good results.

In general, the weight updating strategy employs the correlation between videos and keywords to increase the weights of "better" keywords and to reduce the weights of noisy keywords or even remove them.

Input
$VC = \{vc_1, vc_2, \ldots, vc_k\}$ // video clusters

Output
W = $\{w(t)\}$ //keywords' weights

Initialization
$tf(t) = 0, df(t) = 0 \quad \forall t \in T$

Process
for each video cluster $vc_i \in VC$
  for each keyword t for the video in the cluster vc_i
    if $f(t,vc_i) > thd_1$ $df(t)$++ end if
    if $f(t,vc_i) > tf(t)$ $tf(t) = f(t,vc_i)$ end if
  end for
end for
for each keyword $t \ \forall t \in T$
  if $df(t)$!= 0 $w(t) = tf(t) * \log\frac{|VC|}{df(t)}$
  else $w(t) = 0$ end if
  if $w(t) < thd_2$ remove t end if
end for

Figure 3: The keywords' weight updating strategy

The numbers of video clusters k and the number of keyword clusters l are determined empirically by giving a range and selecting the optimal ones which minimize the mutual information, for the reason that the quality of a co-clustering is judged by the loss in mutual information [3].

Input
C //the co-occurrence table
k // the number of topics (video clusters)
l // the number of keyword clusters

Output
$VC = \{vc_1, vc_2, \ldots, vc_k\}$ // video clusters
$TC = \{tc_1, tc_2, \ldots, tc_l\}$ // keyword clusters

Initialization
$w(t) = 1 \quad \forall t \in T$ //keywords' weights

Process
for n times
  (VC, TC) = Co-Clustering(C, k, l, W)
  W = Weights Updating (VC)
end

Figure 4: Weight-updating co-clustering

4. FINE TOPIC RE-RANKING

4.1 Topic Re-Ranking by Reinforcement Model

The step of weight-updating co-clustering is assigned to coarsely group videos with similar content together so as to filter out topic candidates. As the content of real web videos is abundant, the videos in one cluster would have different relevant scores to the topic. The cluster may even contain some irrelevant videos due to the impact of limited and noisy keywords. Thus, if videos with larger relevant degree can be ranked higher than the ones with smaller relevant degree, it will benefit both users' browsing and websites administrators' organization. It is also very convenient for web service providers to recommend the top N videos to users and remove the last ones. So topic re-ranking could be deemed as a fine topic filtering process as well. On the other hand, if the

corresponding keywords are also ranked depending on their relevant scores, the web administrators can quickly get to know what the topic is about by looking through the top N keywords, which also benefits their organization. Therefore, videos and keywords' ranking is very important for topic discovery.

In this section, we propose a bipartite graph reinforcement model, which ranks the videos and keywords simultaneously based on the analysis of the bipartite graph link structures. The basic idea is that the most relevant keywords are the keywords which are used to annotate many relevant videos and the most relevant videos are the videos which are annotated by the most relevant keywords. This is a mutual reinforcement relationship, which can be represented by an iteration process. Thus, if each video and keyword is assigned a relevant score, their values can be calculated as Equ. (6).

$$\begin{aligned} s_{k+1}(t_i) &= \alpha s_0(t_i) + (1-\alpha) \sum_{v_j \in nb(t_i)} C(v_j, t_i) \times s_k(v_j) \\ s_{k+1}(v_i) &= \beta s_0(v_i) + (1-\beta) \sum_{t_j \in nb(v_i)} C(v_i, t_j) \times s_{k+1}(t_j) \end{aligned} \quad (6)$$

where α, β are the weights ranging from 0 to 1. $s_k(t_i)$ is the relevant score of t_i at iteration times k. $nb(t_i)$ is the neighbor nodes of t_i. Actually, if the videos and keywords' scores are joined to score vectors $SV_k = \{ s_k(v_1), s_k(v_2), \ldots, s_k(v_m)\}$, $ST_k = \{ s_k(t_1), s_k(t_2), \ldots, s_k(t_n)\}$, the reinforcement process can be represented by matrix operation as Equ. (7).

$$\begin{aligned} ST_{k+1} &= \alpha \times ST_0 + (1-\alpha) \times C^T \times SV_k \\ SV_{k+1} &= \beta \times SV_0 + (1-\beta) \times C \times ST_{k+1} \end{aligned} \quad (7)$$

The first row indicates that the videos' relevance propagates to keywords while the second row indicates the keywords' relevance propagates to videos. The reason for updating keywords' relevant scores first is that we consider keywords have more noise than videos.

In this reinforcement process, the initial relevant scores ST_0 and SV_0 are also taken account of. They can be set in two ways: to be the weights obtained from weight-updating co-clustering or to be uniform. The weights α, β indicate how much the initial values are relied on.

4.2 Comparison with HITS Algorithm

HITS (Hypertext Induced Topic Selection) is a link analysis algorithm proposed by Kleinberg [16], which is used to rate web pages. The HITS algorithm first constructs a focus sub-graph which has many relevant pages to a query topic. Then it builds a bipartite graph between the authorities and hubs of the web pages within the sub-graph. It assumes that a good hub is a page that points to many good authorities while a good authority is a page that is pointed to by many good hubs. So the values of hubs and authorities can be calculated as the sum of their neighbors in the other set. And a mutual reinforcement process is used to extract the authorities and hubs iteratively. The work [16] proved that if all the initial values were set to be 1 and the vectors of authorities and hubs were normalized by their 2-norm ($\|*\|_2$) at each iteration, the algorithm would converge.

The main idea of HITS algorithm is similar with our approach of topic discovery. We compare them on the following two aspects.

First, HITS algorithm is a query-dependent link analysis method. It filters out many authoritative pages relevant to the initial query by text-based searching and builds the sub-graph for the specific query topic at first. Our scheme employs weight-updating co-clustering to coarsely group videos and filter out topic candidates as the first step of topic discovery. Essentially, both of the processes aim to get a relative coherent cluster for the next step. However, due to the different applications – searching vs. discovery, the methods are different: HITS uses text-based searching for pre-known queries while our topic discovery is based on co-clustering for the lack of pre-knowledge.

Second, HITS algorithm and our approach both construct a bipartite graph based on the relative coherent cluster. On the bipartite graph, they have similar assumptions: a good node in one set is linked to other good nodes in the other set. Then a mutual reinforcement process is employed to update the scores of nodes iteratively. Essentially, both of the reinforcement processes utilize the inherent tension that exists within the bipartite graph. But our approach is based on the correlation between two modules - videos and keywords while HITS algorithm relies on only one module - the web pages. Besides, our approach takes account of the initial scores, which is not included in HITS algorithm. If we set α, β to be 0 and normalize the score vectors by their 2-norm ($\|*\|_2$) at each iteration time, our reinforcement process will iterate to converge like HITS algorithm. Actually, the initial scores are not very important for topic re-ranking, so they can be removed to assure our reinforcement convergence.

5. TOPIC TRACKING

5.1 Topic Tracking by Reinforcement Model

The traditional topic tracking methods almost rely on one-way representations, which may fail on web videos for two reasons. First, as the keywords are limited and noisy, the one-way representation for web videos is very sparse. Second, as web videos' content is abundant, the videos annotated with other relevant keywords which are different from pre-defined videos may be not found out (as the example in Section 2). In order to overcome these problems, a bipartite graph reinforcement model, which utilizes the bi-direction correlation between videos and keywords, is proposed in this Section for topic tracking.

The main idea is to propagate the relevance of pre-defined videos and keywords to other relevant ones through the links of the bipartite graph, which can be achieved by transferring the relevant scores from one node to its neighbors. So the relevant scores of each video and keyword can be obtained by a mutual reinforcement as in Equ. (8).

$$\begin{aligned} s_{k+1}(t_i) &= \alpha s_0(t_i) + (1-\alpha) \sum_{v_j \in nb(t_i)} p(v_j, t_i) s_k(v_j) \\ s_{k+1}(v_i) &= \beta s_0(v_i) + (1-\beta) \sum_{t_j \in nb(v_i)} p(t_j, v_i) s_k(t_j) \end{aligned} \quad (8)$$

where p is transition probability which is used to avoid explosion of relevance values.

Suppose mq videos and nq keywords are pre-defined to belong to the topic. Their relevant scores as in Equ. (9) are initialized to be non-zero and normalized by its 1-norm ($\|*\|_1$).

$$QSV = \{s(v_1), \ldots, s(v_{mq})\} \quad QST = \{s(t_1), \ldots, s(t_{nq})\} \quad (9)$$

There are other (m-mq) videos and (n-nq) keywords among which the topic is tracked, whose relevant scores are all initialized to be zero.

$$TSV = \{s(v_{mq+1}), \ldots, s(v_m)\} \qquad TST = \{s(t_{nq+1}), \ldots, s(t_n)\} \quad (10)$$

If the score vectors are joined together as in Equ. (11),

$$SV = \{QSV, TSV\} \qquad ST = \{QST, TST\} \tag{11}$$

the reinforcement process can be represented as the matrix operation in Equ. (12).

$$\begin{aligned} ST_{k+1} &= \alpha \times ST_0 + (1-\alpha) \times P_V^{\ T} \times SV_k \\ SV_{k+1} &= \beta \times SV_0 + (1-\beta) \times P_T^{\ T} \times ST_k \end{aligned} \tag{12}$$

where P_V is the transition matrix from videos to keywords while P_T is the transition matrix from keywords to videos. They can be obtained from co-occurrence matrix C by Equ. (13).

$$\begin{aligned} &P_V = D_r^{-1} C \quad P_T = D_c^{-1} C^T \\ &\text{where } D_r(i,i) = \sum_{j=1}^{n} C(i,j) \qquad D_c(i,i) = \sum_{j=1}^{m} C(j,i) \end{aligned} \tag{13}$$

The second row means: D_r and D_c are diagonal matrixes, and their (i; i)-elements equal to the sum of the i-th row and the sum of the i-th column of C respectively. Actually P_V and P_T are normalized matrixes from C. The reinforcement process converges after a few iteration (about 10 in our experiments) [20].

The initial scores are very important for topic tracking, because they are the headstream of the propagation. The weights α, β are often set large to indicate more confidence on initial scores.

The reinforcement process for topic tracking works like random walk [1] [6] [23], which is a kind of methods to rate web pages by estimating how frequently the node will be visited by a random "surfer" on the graph. Random walk assumes that Internet surfers will "random walk" to a web page following the hyperlinks within the current web page, or randomly "jump" to a web page out of the linked set. Random walk with restart (RWR) [6] takes into account the initial scores (the restart node) when ranking the nodes in the graph. Our reinforcement model works like RWR, in which the initial relevant scores are taken account of and the relevance is propagated through the links between videos and keywords iteratively. The main difference is that our model is built on a bipartite graph which indicates the correlation between two modules – videos and keywords.

5.2 Comparison with Topic Re-Ranking

The forms of topic re-ranking and topic tracking processes seem very similar. Both of them are based on the bipartite graph between videos and keywords. And they both have the assumption: a good node in one set is connected with other good nodes in the other set. So they utilize a reinforcement model to update the relevant scores iteratively. The major differences between their reinforcement models are put up on two aspects.

First, co-occurrence matrix is normalized in topic tracking while it is not in topic re-ranking.

Second, although the initial relevant scores are taken account of in both of the reinforcement models, they are essential for topic tracking while they can be removed in topic re-ranking.

In fact, the above two formal differences are related to the essences of the two processes described as below.

First, re-ranking is the second step of topic discovery. Before it, our scheme employs the weight-updating co-clustering to filter out topic candidates first. So re-ranking is performed on a relative coherent cluster like HITS algorithm. Most of the videos and keywords in the cluster are relevant to the same topic. So it can be deemed as a topic-dependant re-ranking. And the original links of the bipartite graph are very important which indicate the inherent focus of the topic, thus they can not be normalized. On the other hand, as most of the videos and keywords are relevant to the topic, the initial relevant scores are not so important and they can be set uniformly. The link structure analysis is used to distinguish the relevant degree without initial impact.

Contrastively, topic tracking is performed on a mixed video data set. Most of the videos and keywords are irrelevant to the tracking topic. So they must have different relevant scores to the topic initially. The main idea of topic tracking is to propagate the relevant scores from pre-defined videos and keywords to other relevant ones. The links between videos and keywords indicate the route of propagation, and in order to avoid explosion of relevance they should be normalized to represent the transition probability.

6. EXPERIMENTS

6.1 Data Set

We obtained 15 days worth of data, 11/1 2006 to 11/15 2006, from YouKu. The data includes more than 20,000 videos (about 200G in file size) and their corresponding metadata[2]. According to the rule of YouKu, a user should annotate several tags and a title when uploading videos. The tags are words while the title is a sentence. We use a natural language processing (NLP) tool [12] to parse the titles and tags. The parsed results compose the keywords set.

We constructed three datasets for different purposes. First, we build a 4-topic data set with ground truth. This data set will be used to compare various proposed approaches and traditional approaches. Second, we use the first 5 days' videos as the data set for topic discovery. We invite 10 participants to do a user study and evaluate the results. Third, we track three most popular topics on the whole 15 days' videos by the bipartite graph reinforcement model, and show interesting observations.

6.2 Effectiveness of Weight-Updating Co-Clustering

In this experiment, we first manually select about 350 videos on four popular topics as shown in Table 3. These videos are selected by our pre-knowledge. For example, we know many super girls' names, nicks and also know the popular "web-terms" relative to the topic. Thus we select the videos annotated by any of these words to compose the video set on the topic "Super-girl". In this way, we can get the ground-truth for the following objective evaluation.

Based on the four topics data set with ground-truth, we aim to evaluate two aspects of coarse topic filtering – the effect of weight-updating strategy and the correlation impact. Thus, we produce 3 clustering results by 3 methods: co-clustering without/with weight-updating and K-Means with weight-updating. The weight-updating K-Means method, which is set to compare with weight-updating co-clustering fairly, means that the videos are clustered by K-Means and the keywords' weights are updated in the same way as in Fig. 3.

[2] The information on privacy has been filtered out first.

Table 3: 4 hot topics

Topic Description	# of Video
Super girl: a very popular annual national singing contest in China for female contestants. The topic includes the contest TV programs, the MTV or news about some super girls, etc	86
Basketball, NBA: this topic includes NBA games, news about NBA players such as Yao Ming, McGrady, Kobe, etc. and other basketball videos	102
Jay Zhou: a popular male singer in Hong Kong. This topic includes his MTV, news about him and his movies	149
Golden mic: a show host contest in a university. This topic includes the introduction to the contest, some students' display, etc	26

Table 4: The matrix of clustering results

Traditional Co-Clustering				Co-Clustering with weight updated twice				K-Means with weight updated twice			
2	**87**	27	0	0	1	0	**26**	0	**70**	0	0
23	14	**23**	3	6	**102**	0	0	0	0	**86**	0
8	0	**94**	0	16	0	**149**	0	24	29	**62**	26
53	1	5	23	**60**	0	0	0	**44**	0	0	0

In all methods, the video cluster number is fixed to 4. Thus a 4*4 matrix can be obtained as shown in Table 4. Each row stands for a video cluster while each column stands for a topic. So the entry (*Cl*, *Tp*) is the number of videos which belong to topic *Tp* in cluster *Cl*. Each row's maximum number is put in bold, which indicates the main topic of the cluster. It's obvious that the weight-updating co-clustering outperforms the other two methods, because 1) its clusters mostly focus on one single topic, and 2) it discovers all the four topics, while the other two methods miss topic 4 due to its small size.

Validating clustering results is a non-trivial task which is also discussed a lot in the work [3]. In our scheme, we evaluate the clustering results through two aspects: 1) one cluster should focus on only one topic; 2) one topic should be concentrated in only one cluster. Thus, considering the former aspect, we first match one cluster to the topic which most of the videos in the cluster belong to. In this way, several clusters may be matched to the same topic. Then considering the latter aspect, from the above matched clusters, we select the one which most of the videos on the topic are concentrated in. Let $A(Cl, Tp)$ be the value of the entry (Cl, Tp) in the matrix above, then the one-one matching from topic to cluster $M(Tp)$ is defined as Equ. (14).

$$M(Cl) = \arg\max_{Tp} A(Cl,Tp) \quad M(Tp) = \arg\max_{Cl:\, M(Cl)=Tp} A(Cl,Tp) \quad (14)$$

Notice that $M(Tp)$ may be null for some topics. After obtaining the one-one matching, we can calculate the precision and recall of the topics to evaluate the clustering results as Equ. (15).

$$P(Tp) = \begin{cases} 0 & M(Tp) = null \\ \dfrac{A(M(Tp),Tp)}{|M(Tp)|} & otherwize \end{cases} \quad (15)$$

$$R(Tp) = \begin{cases} 0 & M(Tp) = null \\ \dfrac{A(M(Tp),Tp)}{|Tp|} & otherwize \end{cases}$$

where |*| is the number of videos in a cluster or topic.

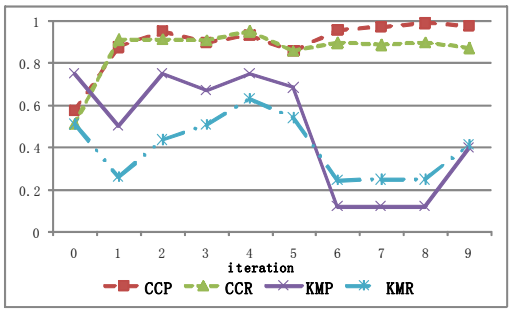

Figure 5: Average precision and recall

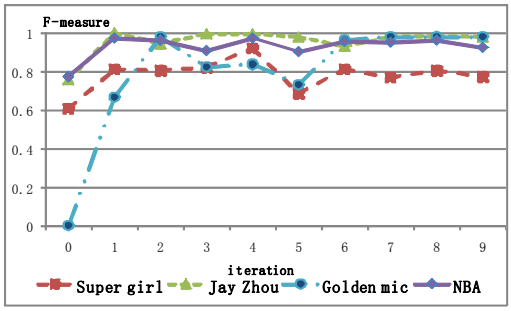

Figure 6: F-measure of weight-updating co-clustering

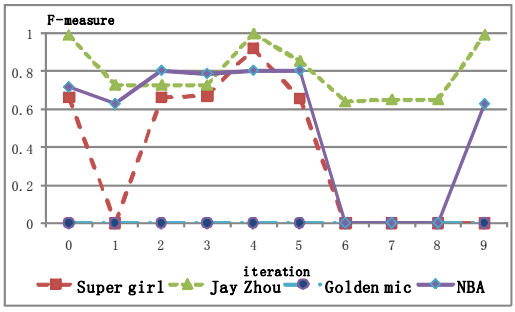

Figure 7: F-Measure of weight-updating K-Means

The curves of the three methods' average precision, recall of the 4 topics are shown in Fig. 5, where CCP, CCR are respective the precision and recall for co-clustering with weight-updating while KMP, KMR are the ones for K-means with weight-updating. In order to consider precision and recall simultaneously, we also use F-measure $F(Tp) = 2 * P(Tp) * R(Tp) / (P(Tp) + R(Tp))$ to evaluate the results. The curves of the 4 topics' F-measure changing with iteration are shown in Fig. 6 and Fig. 7.

When iteration equals to zero, it means no weight updating. So the results show that one or two rounds of weight-updating can improve the performance significantly. On the other hand, the results also demonstrate that weight-updating strategy can improve the performance of co-clustering, but it can not for the performance of K-Means. That is because the co-clustering utilizes the correlation between videos and keywords. The quality of video clusters can therefore be improved by the affinity of keywords clusters when dealing with the problems of limited and noisy textual information.

Table 5 shows the keywords clusters, where B.C. Zhou and L.Y. Zhang are two famous super girls. The table shows that related keywords, e.g., Rockets and Yao Ming, are automatically clustered together by co-clustering.

Table 5: Keyword clusters after weight updating twice

篮球(basketball)
火箭(Rockets); 姚明(Yao Ming)
超级(super); 女生(girl); 唱(sing); 周笔畅(B.C. Zhou); 张靓影(L.Y. Zhang)
NBA
周杰伦(Jay Zhou)
金(golden); 比赛(contest); 介绍(introduction); 主持人(show host); 自我(self); 话筒(mic); 选手(player)
超(super); 号(number); 女声(girl's voice)

6.3 Topic Filtering and Re-Ranking

We do this experiment for two goals. First, we aim to test the performance of coarse topic filtering on complex data set which is composed with many videos on non topics. Second, we want to test the effectiveness of re-ranking method. Thus we use the first 5 days' videos without manually processing as the testing data set.

First, the weight-updating co-clustering method is employed to filter out the topics coarsely. As the coarse filtering step is assigned to provide the "hot" topic candidates, we only select 10 video clusters for evaluation. Video thumbnails are shown in Fig. 8. Each column represents a cluster and each frame represents a video in the cluster.

As whether a video belonging to a topic is more subjective, we conduct a user study to evaluate the performance. Similar to [26], the videos in a cluster are classified to three categories: "Relevant", "Somehow Relevant" and "Irrelevant" to the topic. 10 participants, 7 graduate and 3 undergraduate students, are first asked to review the video clusters, and then give each video a score of 1.0, 0.5, 0.0 to measure its relevant to a topic, which represent "relevant", "somewhat relevant" and "irrelevant" respectively. The relative score of video v is defined as the average score of 10 persons, which is represented as $AS(v)$.

$$AS(v) = \frac{1}{10}\sum_{k=1}^{10} score_k(v) \tag{16}$$

where $score_k(v)$ is the score annotated by person k for video v.

The precision for the video cluster vc is defined as Equ. (17).

$$P(vc) = \frac{\sum_{v \in vc} AS(v)}{|vc|} \tag{17}$$

$|vc|$ is the video number in vc. Table 6 shows the 10 video clusters' sizes and precisions, which demonstrate that some significant topics can be filtered out by weight-updating co-clustering from the real complex data set.

Then the videos and keywords are re-ranked simultaneously by the bipartite graph reinforcement model. The meaningful keywords with the top 5 rank are selected and shown in Table 6, which can explain the topic content.

Table 6: 10 video clusters from coarse filtering

Topic	#of video	Precision	Main keywords
1	151	0.916	刘德华(D. H. Liu)、刘德(D. Liu)、MTV
2	97	0.84	广告(advertisement)、林志玲(L. Z. Lin)、OLAY
3	71	0.877	篮球(basketball)、NBA、火箭(Rockets)
4	61	0.837	劲舞(dance)、舞蹈(dance)、团(club)
5	27	0.807	话筒(mic)、比赛(contest)、主持人(show host)
6	28	0.541	杀(kill)、剪(cut)、鬼子(evil)
7	83	0.828	剧(drama)、艺术(art)
8	98	0.756	绝活(stunt)、魔术(magic)、绝技(stunt)
9	44	0.633	车(car)、卡丁(Kating)、跑跑(running)
10	10	0.46	话筒(mic)、卡丁(Kating)、比赛(contest)

The curves of videos' AS changing with the rank are shown in Fig. 9 (we only choose 2 topics to visualize the curves clearly). It demonstrates that re-ranking step is effective and can rank the more relevant videos higher than the less relevant ones, which matches human's judgments.

The average precision [5] is often used to evaluate the effectiveness of ranking.

$$AP(vc) = \frac{1}{\sum_{i=1}^{|VC|} AS(v_i)} \sum_{i=1}^{|VC|} \left(AS(v_i) * \frac{\sum_{j<i} AS(v_j)}{i} \right) \tag{18}$$

We compare the AP of clusters with random order and the AP after re-ranking in Fig. 10. The results demonstrate that the average precision are improved a lot after re-ranking and except the bad video cluster 10, other clusters' average precision can almost arrive at 90%. So it further demonstrates that the bipartite graph reinforcement model can rank more relevant videos higher than less relevant videos.

6.4 Topic Tracking

This experiment is done to test the performance of topic tracking by bipartite graph reinforcement model. Thus the whole 15 days' data are used, among which three most popular topics on those days are tracked. The three topics are "Super-girl", "basketball" and "Jay Zhou" as described in Table 3. For each topic tracking, we input 10 videos and 5 keywords which are deemed to be relevant to the topic. As the size of testing data set is too large, we only check the top 250 results manually.

Figure 8: Thumbnails of 10 video clusters

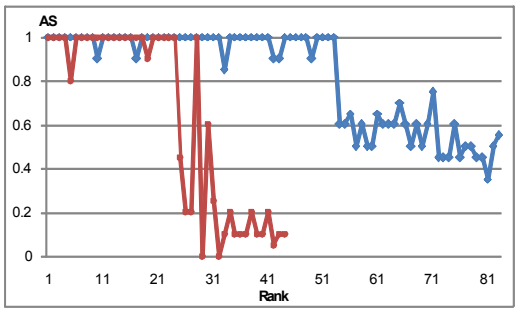

Figure 9: Average scores changing with the rank on two topics

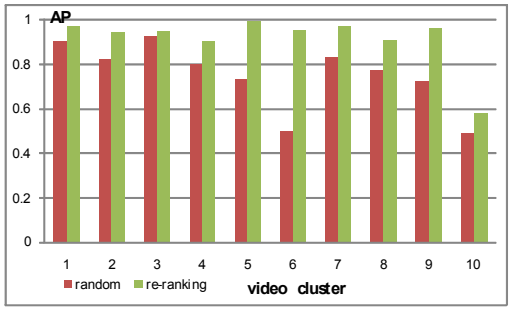

Figure 10: Average precision of the 10 topics

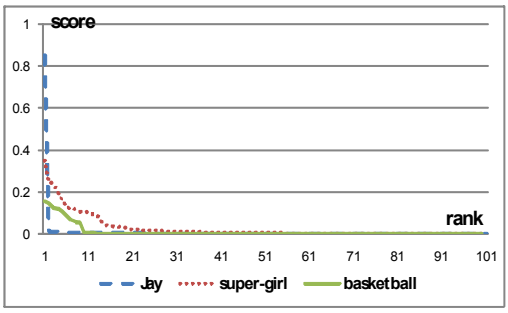

Figure 11: The relevant scores of keywords at top 100

In order to evaluate the bipartite graph reinforcement model's effectiveness, we also employ the traditional tracking method [25], which represent a video as the vector of keywords and use cosine angle to measure videos' relevant degree to the topic.

Table 7 shows the results. M1 and M2 stand for bipartite graph reinforcement model and traditional model respectively. # of video is the number of videos belonging to each topic among the top 250 videos. The results demonstrate that our bipartite graph reinforcement model can find out more videos than traditional model. That's because our model utilizes the correlation between videos and keywords to propagate the textual information and overcome the limited and noisy problems.

In order to evaluate the ranking performance of topic tracking, we also calculate the average precision *AP* of the top 250 videos. The results in Table 7 show that the *AP* of bipartite graph reinforcement model is much higher than the one of traditional model on topic "Super-girl", but they are nearly the same on topic "Jay". It indicates that our bipartite graph model can rank topic "Super-girl" better than the traditional model, but it has almost the same ranking on topic "Jay". The reason for this is related to the topics themselves. "Super-girl" is a very broad topic. It includes many super girls' videos, e.g. Li Yu-chun, Zhang Liang-ying, Zhou Bi-chang etc. Although we don't input these super girls' names, the bipartite graph reinforcement model can utilize the correlation between videos and keywords to propagate the relevant score from pre-defined keywords to these relevant ones. However, the topic "Jay" is relative "concentrative", thus the propagation based on the bipartite graph model doesn't make significant sense on it. Fig. 11 shows the top 100 keywords' relevant scores obtained from bipartite graph reinforcement model. It demonstrates that the major keywords for topic "Super-girl" are much more than the ones for topic "Jay", which also proves that "Super-girl" is a "broader" topic than "Jay". That's why bipartite graph reinforcement model works better on "Super-girl" than on "Jay".

In general, the bipartite graph reinforcement model utilizes the correlation between videos and keywords to propagate the textual information and overcome the problems of limited and noisy keywords, so when tracking broad topics, it outperforms the traditional one-way model greatly, not only on higher recall (more videos will be found out), but also on better ranking results.

Table 7: Topic tracking results

Topic	# of video		*AP*	
	M1	M2	M1	M2
Super-girl	98	73	0.890	0.757
basketball	201	190	0.938	0.911
Jay Zhou	154	151	0.985	0.988

7. CONCLUSION

Most of current solutions of topic detection and tracking were done on news and cannot be directly applied on web videos, because the semantic information of web videos is much less than that of news videos. In this paper, we proposed a bipartite graph model for topic discovery and tracking on web videos. Topic discovery is achieved in two steps – coarse topic filtering and fine topic re-ranking. First, the topic candidates are coarsely filtered by the weight-updating co-clustering algorithm, and then the videos on each topic are re-ranked by analyzing the link structures of the corresponding bipartite graph. Topic tracking is also based on the bipartite graph model, and its main idea is to propagate the relevant scores of pre-defined videos and keywords to other relevant ones through the bipartite graph's links. Both of the re-ranking and tracking are implemented as an iterative reinforcement process. The experiments demonstrate the effectiveness of the bipartite graph. It is a bi-direction model that utilizes the correlation between videos and keywords, and works better than the traditional one-direction models. Our future work will focus on dynamic topic refinement, as well as incorporating online users' relevance feedback.

8. ACKNOWLEDGMENTS

The authors would like to thank YouKu for supplying us with real web videos data. This research is for non-commercial purpose and has been supported by 973 Program under Grant No. 2006CB303103, the National High-Tech Research and Development Plan of China (863) under Grant No. 2006AA01Z118 and the National Natural Science Foundation of China under Grant No. 60573167.

9. REFERENCES

[1] Brin, S. and Page, L., The anatomy of a large-scale hypertextual web search engine. In Proceedings of the Seventh International World Wide Web Conference, 1998.

[2] Dhillon, I.S. et al., Co-clustering documents and words using bipartite spectral graph partitioning, in Proc. 7th ACM KDD'01, pp. 269-274.

[3] Dhillon, I. S. et al., Information theoretic co-clustering, in Proc. 9th ACM SIGKDD'03, pp. 89-98.

[4] Hsu, W. H. and Chang, S.-F., Topic Tracking across Broadcast News Videos with Visual Duplicates and Semantic Concepts, in Proc. Int'l Conf. Image Processing (ICIP), IEEE Press, 2006, pp. 141–144.

[5] Hsu, W. H., Kennedy L. S. and Chang, S.-F., Video Search Reranking through Random Walk over Document-Level Context Graph, in ACM Multimedia, 2007.

[6] Haveliwala, Taher H. Topic-sensitive PageRank. In Proceedings of the Eleven International World Wide Web Conference, 2002.

[7] http://www.youtube.com

[8] http://mashable.com/2006/07/17/youtube-hits-1-million-videos-per-day/

[9] http://www.youku.com

[10] http://www.tudou.com

[11] http://www.6room.com

[12] http://www.nlp.org.cn/

[13] http://www.alexa.com/data/details/traffic_details?url=youku.com

[14] Ide, I., Mo, H., Katayama, N. and S. Satoh, Topic Threading for Structuring a Large-Scale News Video Archive, International Conference on Image and Video Retrieval, 2004.

[15] Kender, J. R. and Naphade, M. R., Visual concepts for news story tracking: analyzing and exploiting the NIST TRECVID video annotation experiment, in CVPR, 2005.

[16] Kleinberg, J.M. Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5), 2000, 604–632.

[17] LDC, TDT3 evaluation specification version 2.7, 1999.

[18] Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V. and Thomas, S. Relevance Models for Topic Detection and Tracking, in Proceedings of the Human Language Technology Conference (HLT), pp.104-110, 2002

[19] Larkey, L. et al., Language-specific Models in Multilingual Topic Tracking, in Proceedings of ACM SIGIR, July 2004.

[20] Langville, A. N. and Meyer, C. D., A survey of eigenvector methods for web information retrieval. SIAM Review, 47(1):135–161, 2005.

[21] Leek, T., Schwartz, R. M., and Sista, S. Probabilistic approaches to topic detection and tracking. In Topic detection and tracking: Event-based information organization, J.Allan (ed.). Boston, MA: Kluwer, 67-83, 2002.

[22] Oard, D. W. Adaptive vector space text filtering for monolingual and cross-language applications. PhD dissertation, University of Maryland, College Park, 1996.

[23] Page, L., Brin, S., Motwani, R., and Winograd, T. The Pagerank Citation Ranking: Bringing Order to the web. Technical report, Stanford University, Stanford, CA, 1998.

[24] STEAMING MEDIA WORLD, Jan, 2007 http://www.lmtw.com/

[25] Yang, Y. et al., Learning approaches for detecting and tracking news events, IEEE Intelligent Systems, vol. 14, no. 4, 1999.

[26] Zhai, Y. and Shah, M., Tracking news stories across different sources, in ACM Multimedia, 2005.