

TOPIC MINING ON WEB-SHARED VIDEOS

Lu Liu¹, Yong Rui², Li-Feng Sun¹, Bo Yang¹, Jianwei Zhang³, Shi-Qiang Yang¹

¹Tsinghua University, China, ²Microsoft China R&D Group, ³University of Hamburg, Germany

ABSTRACT

Internet videos have grown exponentially with the help from video sharing websites. Automatic topic mining is therefore increasingly important for organizing and navigating such large video databases. Most of current solutions of topic detection and mining were done on news videos and cannot be directly applied on web videos, because of their limited and noisy semantic information. In this paper, we will try to address this problem and propose an automatic topic mining framework on web videos. We develop an iterative weight-updated co-clustering scheme to filter “noisy” tags and mine the “hot” topics. We then propose a visual-based clustering approach to further group the videos with similar content, and rank the visual-similar groups by their similarity to the topic center. Experiments on a large web video database demonstrate the superior performance of our weight-updated co-clustering to both of the traditional co-clustering and K-Means. The experiments also demonstrate significant improvement of users’ experience by our visual-based clustering and ranking.

Index Terms— *topic mining, web video, co-clustering*

1. INTRODUCTION

Thanks to the explosive growth of Internet and multimedia technology, online video distribution websites, e.g., YouTube [1], are becoming very popular in recent years. These websites in turn result in a fast expanding web video pool, which makes it very difficult for people to find what they are interested in. Furthermore, website organizers also feel tired to select “hot” topics among the large database to recommend on their front page in order to attract more customers. Thus we are motivated to investigate the problems of mining topics on web-shared videos automatically. We deem videos on the same event or the same person to be of the same topic. For example, TV programs of NBA games, reports on a NBA player and other re-edited videos related to NBA are all videos on the same topic. Automatic topic mining on web videos is an essential task to organize and navigate the large volume of web videos. First, it provides “hot” topic candidates for website organizer, from which they are able to recommend to customers. Second, it benefits users’ browsing and search experience as they can trace back in time on what they are interested in, and also subscribe to a topic to get future updates. Third, topic mining is also useful for advertisers to analyze the interest of the masses and choose an attracting type for advertisement.

Since NIST first proposed the problem of Topic Detection and Tracking (TDT) in the 1990s [3], a lot of work has been done on news documents and videos. Yang et al. [4] represented news documents as vectors of words and used cosine angle to measure their similarity, while Larkey et al. [5] estimated the language model of documents and measure the similarity by the symmetric clarity-adjusted divergence. Kender et al. [6] incorporated both high-level semantic features and temporal properties to measure

the video similarity and used the normalized cut to cluster the news stories. Hsu et al. [7] developed a multimodal fusion framework for topic tracking by incorporating low-level features, visual near-duplicates, cue words and semantic concepts. Zhai et al. [8] combined both the visual content and spoken language content in videos to link news stories on the same day.

Comparing to news, web videos have very different properties. First, the semantic information of web videos is both limited and noisy. Most of web videos only have a title and a few tags. And their content is difficult to be recognized by automatic speech recognition (ASR). Furthermore, similar content may be annotated differently by different people. Even worse, sometimes irrelevant but “popular” tags are used to artificially boost the video rank. Second, the temporal property of web videos is not as obvious as news. The news stories on the same topic will appear in nearby days. But the duration for web videos’ topic may last as long as several months depending on the interest of the mass.

In this paper, we propose a framework for topic mining on web videos. In order to utilize the correlation between videos and tags and overcome the problem of limited and noisy text information, we develop an iterative weight-updated co-clustering scheme to cluster videos as well as tags simultaneously. Then we propose a visual-based clustering approach to further group the videos with similar content, and rank the visual-similar groups by their similarity to the topic center. We conduct extensive experiments on a large web video dataset supplied by Yoqoo [2], one of the most popular video sharing websites in China, and report very promising performance.

2. PROPOSED FRAMEWORK

2.1. Topic Discovery by Co-Clustering

Table 1: The tags of 4 videos on the topic “NBA”¹

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

Topic is a high-level semantic concept, which indicates the common focus of a group of videos. Topic mining is an automatic process to obtain clusters in which videos with similar semantic content are grouped together. The tags of web videos are provided by video owners. Since different people may use different words to describe the same topic, videos on the same topic may not have the same tags at all. Thus, directly clustering videos will fail for diverse web videos. Fortunately, we observe that tags related to the same topic co-occur frequently in large number of web videos. This implies that there also exist tag clusters as well as video clusters. Thus videos on the same topics but with different tags can be clustered together by the affinity of tag clusters. Table 1

¹ We translated all Chinese tags into English in this paper.

gives an example, from which the tag cluster “Rockets; Yao Ming; NBA; basketball” can be obtained easily. Thus, although Video 2 and Video 4 only have one common tag, they can still be clustered together by the assistance of the tag cluster above. In addition, tag clusters also help to reduce the influence of inexact tags, such as “newsreel” in Video 3.

As the clustering processes of videos and tags are inherently dependent on each other, we adopt the information-theoretic co-clustering algorithm [9], which uses the correlation between videos and tags to cluster them simultaneously.

First, the titles are parsed by a natural language processing (NLP) [10] tool. The tags and the participle results of titles are used as videos’ keywords. Then the keywords are filtered through two processes – word type filtering and mutual information filtering. The former process is to filter out the stop words and other words with ambiguous word types such as adjective, adverb, etc. And the latter process is to filter out the keywords with small information, which is measured similar to [7] as below:

$$IE(k) = p(k) \sum_v p(v|k) \log \frac{p(v|k)}{P(v)} \quad (1)$$

where k 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.

Although the above two processes can remove a lot of noisy keywords, the left keywords still have different impacts on topic mining. On the other hand, some web videos may not belong to any topic, thus they should be also removed before clustering. Based on these observations, we assign each keyword and video a weight and develop an iterative weight-updated co-clustering method to remove “noise” and improve the impact of more significant tags.

The traditional information-theoretic co-clustering algorithm requires inputting the joint probability distribution matrix C [9] between videos and keywords. And it outputs the video clusters $VC = \{vc_1, \dots, vc_{|VC|}\}$ and keyword clusters $KC = \{kc_1, \dots, kc_{|KC|}\}$ simultaneously. The correlation between videos and keywords is used to update their weights based on the clustering results. Suppose $F(k, vc_i)$ represents the frequency of keyword k in video cluster vc_i .

$$F(k, vc_i) = \frac{N(k, vc_i)}{|vc_i|} \quad \text{where} \quad N(k, vc_i) = \sum_{v \in vc_i} I(k, v) \quad (2)$$

where $I(k, v)$ equals to 1 if k occurs in the video v , 0 otherwise. $|vc_i|$ is the sum of all the keywords’ occurrence in vc_i . Let $tf(k)$ denotes the maximal frequency of k , and $df(k)$ denotes the number of video clusters k appears. Then the keywords’ weights are updated as in Fig.1.

The main idea of this weight updating process 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 too common and should be assigned a smaller weight. Besides, the threshold thd_1 is set for reducing noise. The video weights update based on the tag clusters in the same way as keywords.

After weight updated, the input matrix is modified by multiplying the weight values as below:

$$C(i, j) = C(i, j) * W(i) * W(j) \quad (3)$$

And then the modified matrix is input and the videos and keywords are co-clustered again. This process is repeated until the average similarity AS of video clusters is larger than a threshold.

$$AS = \sum_{i=0}^{|VC|} \sum_{v \in vc_i} Sim_{\cos}(v, vc_i) \quad (4)$$

where $Sim(v, vc_i)$ is the cosine distance in Equation (7) which will be explained in Section 2.2. AS is used to judge the quality of clustering.

In general, our approach assures that “better” keywords and videos have more impact by Equation (3), and the impact of noise is reduced by removing videos and keywords with small weights.

```

tf(k) = 0, df(k) = 0
for each video cluster vc_i in VC
    for each keyword k for the video in the cluster
        if F(k,vc_i) > thd_1      df(k)++      end if
        if F(k,vc_i) > tf(k)      tf(k) = F(k,vc_i)  end if
    end for
end for
for each keyword k
    if df(k) != 0
        W(k) = tf(k) * log(|VC| / df(k))
    else
        W(k) = 0      end if
    if W(k) < thd_2      remove k      end if
end for

```

Fig 1: The keywords weight updating strategy

Besides the matrix C , the co-clustering algorithm also requires inputting the cluster numbers of videos and tags. Without prior knowledge, our approach gets the cluster numbers 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 [9].

2.2. Visual Clustering and Ranking

After topic mining, the videos are in random order in the cluster. The videos are therefore needed to be ranked to improve the users’ experience. The similarity between a video and a cluster could be measured based on textual information. But such approach will have poor results for two reasons. First, there are few tags in a video, which are unstable for measuring similarity between the video and the cluster. Second, videos with high visual similarity but different keywords will be ranked quite differently, which are inconsistent with humans’ perception.

Thus, in our approach, sub-clusters are first made within each cluster, in which videos are visually similar. Then sub-clusters are ranked using textual information, but the ranking orders within a sub-cluster are no longer important. This is based on the assumption that visually similar videos are mentioning the same topic; thus, their orders are useless. Such assumption indicates that our visual measuring method must be quite robust. Considering precision requirement and efficiency, we adopt the duplicate detection method [11], and measure the visual similarity of two videos v_1 and v_2 as below

$$Sim(v_1, v_2) = DFN / (FN(v_1) + FN(v_2)) \quad (5)$$

DFN is the number of duplicate key frames between v_1 and v_2 , $FN(v_i)$ is the number of key frames in v_i . A graph-connecting method is employed to cluster the videos with the similarity more than a threshold.

After obtaining sub-clusters, their similarities with the corresponding topic can be measured by their textual information.

The topic model of video cluster vc_i is defined as the frequency vector on keywords,

$$TM(vc_i) = \{F(k_1, vc_i), F(k_2, vc_i), \dots, F(k_n, vc_i)\} \quad (6)$$

where n is the number of keywords, The sub-clusters are represented in the same way. Then the weighted-cosine distance is used to measure the similarity between the two vector $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$

$$Sim_{\cos}(A, B) = \frac{\sum_i W^2(k_i) a_i b_i}{\sqrt{(\sum_i W^2(k_i) a_i^2)(\sum_i W^2(k_i) b_i^2)}} \quad (7)$$

The keywords are weighted by term-frequency inverse-document-frequency (TF-IDF). Based on the similarity, the visual-similar sub-clusters on the topic are ranked.

3. EXPERIMENTS

We use a collection of 5 days' data from Yoqoo [2], YouTube's counterpart in China. The data includes more than 8,000 videos (about 60G in file size) and their corresponding metadata. The metadata contains videos' title and tags annotated by end users. We design two experiments to evaluate different aspects of our proposed framework. First, we use a dataset with ground-truth to conduct objective evaluation of our approach and compare it with the traditional co-clustering and K-Means. Second, we mine the topics on all the videos of the 5 days and conduct a user study to evaluate people's video browsing experience.

3.1. Performance and Comparison

Four popular topics are used in this experiment (see Table 2).

Table 2: 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 3: 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

To test the effectiveness of co-clustering and weight updating algorithms, we produce 3 clustering results by 3 schemes: traditional co-clustering, co-clustering with weight-updated, and K-Means with weight-updated, which means the videos are clustered by K-Means and the keywords' weights are updated in the same way as in Fig.1. The weight-updated K-Means is set to

compare with weight-updated co-clustering fairly. In all schemes, the video cluster number is fixed to 4. Thus a 4*4 matrix can be obtained as shown in Table 4. Each entry (c, t) is the number of videos which belong to topic t in cluster c . Each row's maximum number is put in bold, which indicates the main topic of a cluster. The proposed algorithm outperforms both the traditional co-clustering and the weighted-updated K-Means, because 1) its clusters mostly focus on one single topic, and 2) it discovers all four topics, while the other approaches miss topic 3 due to its small topic size.

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. Because the one-one matching from topic to cluster is not clear, validating clustering results is a non-trivial task. 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 one 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 $A(c, t)$ be the value of the entry (c, t) in the matrix above, then the one-one matching from topic to cluster $M(t)$ is defined as below:

$$M(c) = \arg \max_t A(c, t) \quad M(t) = \arg \max_{c: M(c)=t} A(c, t) \quad (8)$$

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

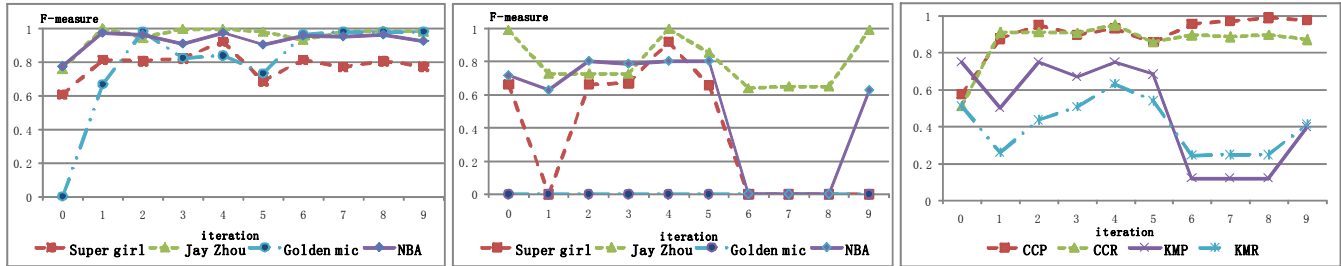
$$P(t) = \begin{cases} 0 & M(t) = null \\ \frac{A(M(t), t)}{|M(t)|} & otherwise \end{cases} \quad R(t) = \begin{cases} 0 & M(t) = null \\ \frac{A(M(t), t)}{|t|} & otherwise \end{cases} \quad (9)$$

where $|*|$ is the number of videos in a cluster or topic. F-measure $F(t) = 2 * P(t) * R(t) / (P(t) + R(t))$. The curves of the 4 topics' F-measure changing with iteration are shown in Fig.2 (a) (b). The curves of different schemes' average precision, recall of the 4 topics are shown in Fig.2 (c), where *CCP*, *CCR*, *KMP*, *KMR* are respective the precision and recall of co-clustering and K-means with weight-updated. The results show that weight-updated co-clustering can improve the performance, while the weight-updated K-Means can not. 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 tag clusters when dealing with limited and noisy text information. Moreover, the performance of our weight-updated strategy outperforms that of the traditional co-clustering without weight-updated significantly for the reason that weight-updated strategy reduces the impact of "noise" during the iteration.

Table 4: Tag clusters after weight update twice

basketball
Rockets; Yao Ming
super; girl; sing; B.C. Zhou; L.Y. Zhang
NBA
Jay Zhou
golden; contest; introduction; show host; self; mic; player
super; number; girl's voice

Besides the video clusters, the tag clusters are obtained at the same time as shown in Table 4, where B.C. Zhou and L.Y. Zhang are two famous Super Girls. The table shows that related tags, e.g., Rockets and Yao Ming, are automatically mined and clustered together by co-clustering.



(a) F-measure of weight-updated Co-Clustering (b) F-measure of weight-updated K-Means (c) Average precision and recall

Fig 2: The curves of F-measure, average precision and recall changing with iteration



Fig 3: Thumbnails of 10 video clusters

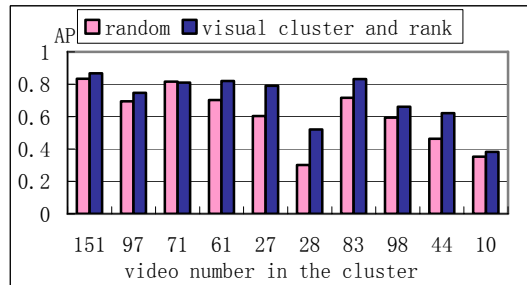


Fig 4: The AP of 10 video clusters

3.2. User Evaluation

In this sub-section, we conduct subjective user studies to evaluate end-user’s video browsing experience. We report experimental results on the whole 5 days’ data. Similar to [8], we concentrate on the precision measure. We invited 10 participants, 7 graduate and 3 undergraduate students, to attend a user study. The participants are first asked to review the video clusters, and then asked to 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 precision for the video cluster vc is defined as

$$P(vc) = \frac{1}{10} \sum_{i=1}^{10} \frac{\sum_{v \in vc} score_i(v)}{|vc|} \quad (10)$$

where the $score_i(v)$ is the score annotated by person i for video v . As the video clusters are assigned to provide the “hot” topic candidates, we only select 10 video clusters for evaluation. Video thumbnails are shown in Fig.3. Each column represents a cluster and each frame represents a video in the cluster. The precisions of the 10 video clusters are shown in Table 5. The first and third rows show the number of videos in each cluster.

We define a modified average precision measurement to evaluate the visual clustering and re-ranking performance.

$$AP(vc) = \sum_{k=1}^{10} \frac{1}{|vc|} \sum_{i=1}^{|vc|} \sum_{j < i} \frac{score_k(v_j)}{i} \quad (11)$$

We compute the AP of clusters with random order and the AP of clusters with further clustered and ranked as described in Section 2.2. The results shown in Fig.4 demonstrate that the visual clustering and ranking significantly improve the user experience a lot.

Table 5: The precision of 10 video clusters

# of Video	151	97	71	61	27
precision	91.56%	83.33%	87.73%	86.64%	80.74%
# of Video	28	83	98	44	10
precision	54.09%	82.79%	75.56%	63.31%	46%

4. CONCLUSION

In this paper, we propose a novel topic mining framework on web videos. We show that how co-clustering algorithm is leveraged to iteratively refine the video clusters with the assistance of keyword clusters. Furthermore, based on duplicated frames, we generate sub-clusters within each cluster, and then rank them on overall textual information. Our experiments demonstrate that the proposed weight-updated co-clustering outperforms both the traditional co-clustering and K-Means. The results of user study show that the visual clustering and ranking improves the users’ experience significantly. For future work, we will explore dynamic topic refinement with the increase of videos, as well as incorporating online users’ relevance feedback.

5. REFERENCES

- [1] <http://www.youtube.com>
- [2] <http://www.yoqoo.com>
- [3] LDC, “TDT3 evaluation specification version 2.7,” 1999.
- [4] Y. Yang *et al.*, “Learning approaches for detecting and tracking news events,” *IEEE Intelligent Systems*, vol. 14, no. 4, 1999.
- [5] L. Larkey *et al.*, “Language-specific Models in Multilingual Topic Tracking,” in *Proceedings of ACM SIGIR*, July 2004.
- [6] J. Kender *et al.*, “Visual concepts for news story tracking: analyzing and exploiting the NIST TRECVID video annotation experiment,” in *CVPR*, 2005.
- [7] W. H. Hsu *et al.*, “Topic Tracking across Broadcast News Videos with Visual Duplicates and Semantic Concepts,” in *ICIP*, 2006.
- [8] Y. Zhai and M. Shah, “Tracking news stories across different sources,” in *ACM Multimedia*, 2005.
- [9] I. S. Dhillon, *et al.*, “Information theoretic co-clustering,” in *ACM SIGKDD*, 2003
- [10] <http://www.nlp.org.cn/>
- [11] B. Wang *et al.*, “Large-Scale Duplicate Detection For Web Image Search,” in *ICME*, 2006.