Video Highlight Shot Extraction with Time-Sync Comment

Yikun Xian, Jiangfeng Li[*] , Chenxi Zhang, Zhenyu Liao
School of Software Engineering, Tongji University, Shanghai, China
siriusxyk@gmail.com, lijf@tongji.edu.cn, xzhang2000@163.com,
102982liaozy@tongji.edu.cn

ABSTRACT

Benefit from abundance of mobile applications, portability of large-screen mobile devices and accessibility of media resources, users nowadays much more prefer to watch videos on their mobiles no matter whether they are at home or on the way. However, constrained by available time and network flow, users may only choose to watch some hot video segments that are manually annotated by video editors. In this paper, we aim to automatically extract video highlight shot with the help of video sentimental feature of time-sync comments. First, analyzing statistical feature of real data. After, we simulate the generation process of time-sync comment after. Then, we propose a shot boundary detection method to extract highlight shot, which is proved to be more effective than traditional methods based on comment density. This experiment attests the time-sync comment is particularly suitable for sentiment-based video segment extraction for 2 reasons. 1) Text-based similarity calculation of is much faster than image-based process depending on every frame of video; 2) Time-sync comment reflects user subjective emotion therefore is useful in personalised video recommendation.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Data Mining

Keywords

Video highlight extraction; topic model; time-sync comment

1. INTRODUCTION

As functionality of smart phones becomes more powerful and abundant, people nowadays tend to watch videos from YouTube and Hulu on their phones, tablets and other mobile devices. In one scenario, for instance, waiting in the bus station, people would like to watch some hot video segments rather than the whole video that lasts for hours due to limitation of available time, network flow and bandwidth. Currently, these video segments are tagged and annotated manually by video editors, and it definitely costs a lot of time and efforts for the fact that they have to inspect every complete video hours by hours. In other cases, users usually want to look for some similar video shots to what they have previously watched. The word 'similar' here means there are some common sentimental features like exciting, horrible and funny elements in among candidate video shots. For example, a user is more willing to find some other similar funny video segments after he watched a comedy show in Saturday Night Live. Therefore, our goal is to extract potential highlight video shots with textual annotations or tags.

To solve this problem, two groups of existing video processing techniques can be used: one is based on image processing including technologies of shot boundary detection, key frame extraction and scene segmentation, etc.; and the other utilizes external textual information, such as title, author and comments, to annotate and index videos. These techniques indeed perform well in solving problems of whole video processing, but they perform far from satisfaction in video highlight shot extraction due to following reasons. First, image itself can hardly be fully understood by computers as its latent content behind story or plot is unable to be directly retrieved from pixels, unlike what human can easily do. Second, query for video from user input is usually in the form of text string, which is the common and convenient way for users to convey information to computer. So this inevitably leads to a conversion between texts and images, which is another hard and complex topic in image processing. Third, textual information like metadata (especially title and tags) and normal comments can simply reflect upon features of a whole video, but fails to extract a certain segment from the video.

A novel kind of video available both online and in mobile application is burgeoning and becomes increasingly popular. Nico Nico Douga [1] in Japan and bilibili [2] in China are two typical examples. Videos on these sites consist of external textual information known as time-sync comments (or simply TSC, the term first introduced in [13]) or barrages (in [7]). An example of TSC [3] is shown in Figure 1. When a user is watching a wonderful and marvellous scene or story, he will probably post TSC to share his feelings. Different from traditional comments that usually follow a post or a video,

[*]Corresponding author

HotPOST'15, June 22, 2015, Hangzhou, China.
Copyright © 2015 ACM 978-1-4503-3517-1/15/06 ...$15.00.
http://dx.doi.org/10.1145/2757513.2757516.

[1] http://www.nicovideo.jp/

[2] http://www.bilibili.com/

[3] http://www.bilibili.com/video/av2094850/index_2.html

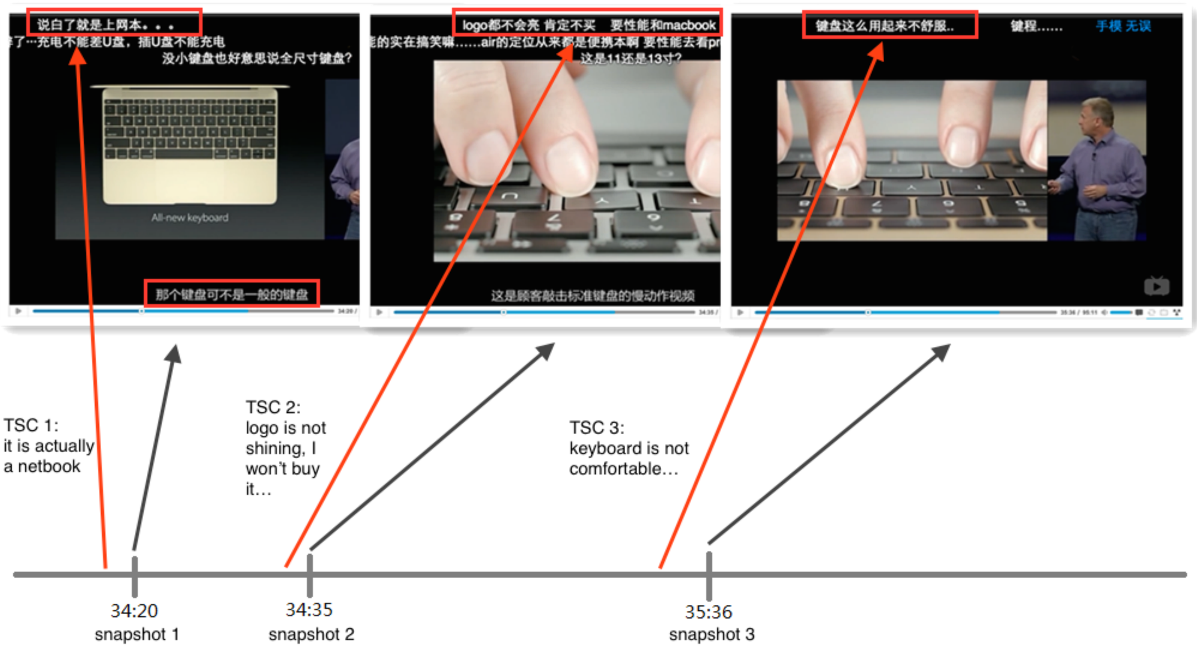


Figure 1: Example of a TSC video about introduction to new Macbook.

TSC simultaneously floats across or fixates on the video and lasts for a period (default 5 seconds on bilibili). Since TSC is generated subjectively by users and reflects their emotions triggered by the specific content in the video, we can apply it to video highlight shot extraction without video or image processing techniques. More importantly, image-based technique for video extraction must involve large amount of images (frames) during phases like feature extraction, similarity calculation and boundary detection, which definitely is more time-consuming than text-based techniques.

To the best of our knowledge, limited work has been done with the research on TSC [7, 13, 14], let alone its application on video extraction. In this paper, we aim to extract the latent video highlight shots that are similar to the given video with the help of TSC. In particular, we first find global sentimental features from all TSC through topic model so that similarity among highlights can be calculated by these features. Topics are represented by sentimental keywords converted from TSC. Then, we segment each video into small basic units, each of which is also represented by sentimental keywords. After, we propose an algorithm for highlight shot boundary detection using basic units and sentimental features. Finally, we simulate the generative process of TSC to generate a large data set for testing the above method. The evaluation index of matching rate shows that our method performs quite well in highlight extraction through TSCs.

2. RELATED WORK

Video Indexing and Retrieval According to the survey [9], the whole process of video retrieval is divided into four parts: structure analysis (shot boundary detection, key frame extraction and scene segmentation), feature extraction, semantic indexing (video data mining and annotation), and query and retrieval. What we target for in this paper is the final step where user provides input query and return similar video highlights (segments). There are four types of information to be used in video [12, 3]: 1) video metadata including title, summary, tags and other basic video information; 2) transcripts and captions text; 3) audio information; 4) visual information in images. Regardless of 2 and 3, text information like title and keywords and visual information like the video segments user has just watched can be regarded as the input of query. Aytar et al. [1] utilize semantic similarity between words to retrieve and rank relevant videos considering to a search query in natural language English. However, it mainly focus on the difficulty of analysing the semantics in natural language and directly uses trained concept detector, which is quite different from our method.

Time-Sync Commented Video TSC is first introduced in [7] to analyze network community. [13] first use TSC to tag selected video shots through topic models. His work mainly focus on how to annotate a given shot by learning existing ones. The given shot is approximately chosen through frequency by peakfinder [4]. , these shots are not always highlight ones because the semantic feature of each highlight shot is missing.

Topic Modeling LDA, namely Latent Dirichlet Allocation, [2, 8] is a generative model for extracting latent topic distribution from documents and has been widely used in text mining or even other irrelevant fields. Two latent variables $\Theta = \{\vec{\theta}_m\}_{m=1}^{M}$ and $\Phi = \{\vec{\varphi}_k\}_{k=1}^{K}$ are respectively the basis for latent-semantic representation of documents and words and inference of these two parameters is actually the core of LDA. Here, we simply refer to Gibbs sampling for parameter inference as it will be used in later section.

[4] http://www.mathworks.com/matlabcentral/fileexchange/25500-peakfinder

Table 1: Notation used in the model

SYMBOL	DESCRIPTION
v	time-sync commented video
T_v	video length (time duration)
N_v	total number of frames in video v
$f_{v,i}$	i-th frame in video v
T_f	frame length (time span)
s_v	highlight shot in video v
$c_{v,i}$	i-th TSC in video v
w	sentimental keyword of TSC
t_0	start time stamp of TSC
T_c	TSC length (time span)
M_v	total number of TSCs in video v
e_v	global emotional feature of video v
$\mathcal{V}$	video corpora of size $\lvert\mathcal{V}\rvert$
$\mathcal{C}$	TSC corpora of size $\lvert\mathcal{V}\rvert$
$\mathcal{W}$	sentimental keywords dictionary of size $\lvert\mathcal{W}\rvert$

3. HIGHLIGHT SHOT EXTRACTION

To simplify the problem, we assume that textual content of each TSC can be fully represented by a keyword coming from a dictionary that covers enough sentimental words. This is because: 1) Intuitively, each user will be triggered at most one feeling after watching a highlight shot that is exciting, horrible or funny, etc. Thus, content of TSC should be monotonous; 2) Statistically, we find that TSC is typical short text with average 6.28 Chinese words (including function words) in our dataset. For more detail on TSC data, please refer to the experiment section.

Besides, we put aside sentimental analysis that mainly focus on converting textual comment to single word because it is not our concentration in this paper and can be directly derived from some existing work [5, 4, 11].

3.1 Problem Definition

A video v of T_v seconds can be segmented into a continuous sequence of frames $v = \{f_{v,1}, f_{v,2}, ..., f_{v,N_v}\}$, where each frame $f_{v,i}$ $(1 \leq i \leq N_v)$ has fixed length T_f, for example, 256ms in most cases. Obviously, count of frames in video v has $N_v = T_v/T_f$, where length of last frame may be shorter than T_f. Similarly, highlight shot s_v is a continuous subsequence of video v, and denoted as $s_v = \{f_{v,i}, f_{v,i+1}, ..., f_{v,j}\}$ $(1 \leq i \leq j \leq N_v)$. As previously assumed, TSC in this paper is a tuple with three elements, namely $c_{v,i} = (w, t_0, T_c)$, where w_k is the sentimental keyword from dictionary $\mathcal{W}$, t_0 is the start timestamp in video v and T_c is the time duration when TSC is floating on screen (default 5 seconds on bilibili). A video usually has many TSCs of different keywords and start timestamps. The set of TSC of video v is denoted as $\vec{c_v} = \{c_{v,1}, c_{v,2}, ..., c_{v,M_v}\}$. Notations used in this paper are listed in Table 1. Relationship between video, shot, frame and TSC is shown in Figure 2.

In this paper, we mainly focus on the problem of highlight shot extraction and retrieval. Formally, for extraction problem, given video v and its TSC $\vec{c_v}$, we want to find possible $\vec{s_v} = \{s_{v,1}, s_{v,2}, ...\}$, where any two $s_{v,i}$ and $s_{v,j}$ are not overlapped (no common frames). For retrieval problem, given query highlight shot s_q and video v with TSC $\vec{c_v}$, we are aimed to extract the highlight shots $\vec{s_v}$ from v so that all highlight shots have similar sentiment. In order to solve above problems, we have to figure out 1) how to make full use of TSC to descibe frame, shot and video, 2) how to calculate similarity with the descriptive model between frames and shots respectively and 3) how to detect the highlight shot boundary (start and end time) in a video.

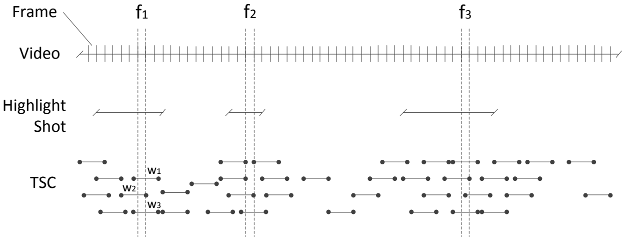


Figure 2: Relationship between video, shot, frame and TSC. Frame is basic element for highlight and video. TSC is external descriptive information for video and frame.

3.2 TSC Features

We firstly discuss some features of TSC, which is useful in frame similarity calculation. TSC is previously defined as a three-element tuple $c = (w, t_0, T_c)$. Suppose that sentimental keyword w has been converted from original comment text through certain mature technologies. Empirically, we observe that user will publish comment only when he is moved or excited by current shot or close preceding shots. This leads to the hypothesis that TSC is much more effective and representative at the beginning priod of $[t_0, t_0 + T_c]$ than at the later one. So, we propose that the strength of TSC f follows the decay function g_c in $[t_0, t_0 + T_c]$:

$$g_c(t) = \begin{cases} T_c^{t_0 - t}, & \text{if } t_0 \leq t \leq t_0 + T_c \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

As we can see, the strongest effect of TSC equals to 1 when it is just published at t_0, and after T_c seconds pass, strength of the TSC becomes almost 0. This obviously explains how TSC can reflect upon user sentiment corresponding to different timestamp.

3.3 Frame Representation and Similarity

As Figure 2 shows, each frame may contain some TSCs, so the easiest way to descibe a frame is the vector of keyword count, $f = (n_1, n_2, ..., n_{|\mathcal{W}|})$, where $n_i(1 \leq i \leq |\mathcal{W}|)$ is the count of i-th keyword in the dictionary. For example, in the interval of frame f_1, there are 3 keywords, $w1$, $w2$ and $w3$, thus, $f_1 = (1, 1, 1, 0, 0, ...)$ and Jaccard similarity [10] can be applied to calculate pairwise frames similarity. It works well when adjacent frames have similar keywords, but fails on two frames which have different keywords with same meaning (synonym). Meanwhile, according to TSC strength decay function, keyword strength is a better choice than keyword count. On the other hand, number of TSCs in the frame interval is much smaller than M_v, which may cause ambiguity of frame meaning.

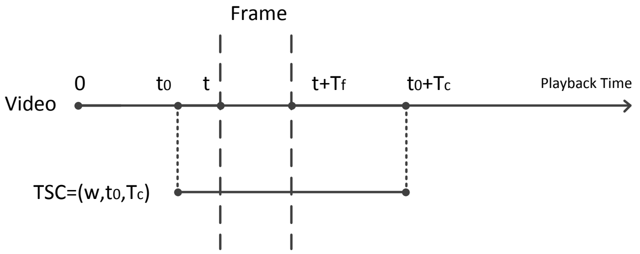


Figure 3: For each dimension for keyword strength vector of the frame, find all TSCs such that time duration interval of each TSC overlaps with that of the frame. Then, calculate component weight for the vector according to Equation 2.

Therefore, each dimension in the keyword strength vector of frame $f_v = (g_1, g_2, ..., g_{|\mathcal{W}|})$ at time t in video v can be calculated as follows:

$$g_i = \sum_{c \in \vec{c_f}} g_c(t) \qquad (2)$$

where $\vec{c_f}$ is the subset of TSCs in video v:

$$\vec{c_f} = \{c | w = w_i, t - T_c < t_0 < t + T_f\}$$

Figure 3 calculating the keyword strength vector of frame.

However, with the keyword strength vector of the frame, it is still inaccruate to measure Jaccard similarity for pairwise frames due to scacity of TSC in each frame. To solve this problem, we introduce a new variable e_v, the global emotional feature of video v. In real cases, the global emotional feature of video corresponds to the feeling expressed by the user who just watches a highlight shot. Meanwhile, local emotional features in highlight shots somehow compose the global emotional feature of video and highlight shot consists of set of frames. In the light of this reason, we calculate the feature similarity between frame and global emotional feature of video rather than the pairwise similarity between frames.

To obtain the global emotional feature of video v, we refer to classic topic model LDA which aims to calculate latent topic variables in documents. Similar to the document-topic-word model, three layers correspond to video-feature-TSC, where keyword in TSC is identical to word in document. On the other hand, the video here simply consists of textual keywords, ignoring visual information, so it can be represented as a weighted combination of global emotional features. Analogous to the topic of a document, global emotional feature is a latent variable defined as the distribution over keywords, $e_v = (n_1, n_2, ..., n_{|\mathcal{W}|})$. To simplify the question, we suppose there are only K unique features in video corpora.

Before applying LDA to feature inference, we point out two minor differences between the generative process of video and that of document.

- **Highlight Shot Exclusion** Highlight shot is an additional and special element in videos while there is no identical concept in documents. As previously said, generation of TSC is largely affected by highlight shot, or more specifically, local emotional feature of highlight shot. In order to simplify the problem, we intentionally regard the generation of TSC is, by and large, influnced by video, namely, the global emotional feature of video. In the experiment, we show that this approximate calculation also performs much better than other traditional methods.
- **TSC Density** Density of TSC mirrors the special characteristcs of TSC video because some existing work use frequency of TSC to extract shots [13]. Unlike document where every word is sequentially generated and uniformly located, the uncertainty whether comment is generated at the time depends on the probability whether user feeling will be triggered by highlight shot. For example, it is more likely to generate comments at the time interval of highlight shots than those out of highlight shots.

Now, we measure the similarity between frames and global emotional features of video. Given video set $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$, and TSC set $\mathcal{C} = \{\vec{c_{v_1}}\}$ where keywords come from dictionary of size $|\mathcal{W}|$, we can get video-TSC matrix of size $|\mathcal{V}|*|\mathcal{W}|$ where element value is the keyword count of the video. In order to decompose the video-TSC matrix into two matrices, namely video-feature matrix of size $|\mathcal{V}|*K$ and feature-TSC matrix of size $K*|\mathcal{W}|$, we apply LDA in this case instead of SVD[6] because LDA performs better on topic analysis despite of its slower inference process than SVD. After that, we can get feature similarity d_f of frame f_v given e_v by cosine similarity between the keyword strength vector of f_v and global emotional feature e_v:

$$d_{f_v} = cos(f_v, e_v) = \frac{\sum_{i=1}^{|\mathcal{W}|} g_i \times n_i}{\sqrt{\sum(g_i)^2} \times \sqrt{\sum(n_i)^2}} \qquad (3)$$

3.4 Highlight Shot Boundary Detection

After we get feature similarity d_f of each frame according to the given global emotional feature of video, we can then detect the range of highlight shots in the video through Algorithm 1. It shows that, during each iteration, the frame with maximum feature similarity for given global emotional feature is first selected, and compared sequentially from middle to two sides. Besides, two thresholds are required to be set. $epsilon$ controls the lower boundary for maximum similarity feature $d_{f_{v,k}}$ of video v for given emotional feature $e_{v,j}$. This is because if $d_{fv,k}$ is too small, there may be no related highlight shot with this emotional feature exists, so we should ignore this feature. δ controls the leftmost and rightmost frames to set boundary for latent highlight shot. A better way is to set dynamic δ according to different videos because threshold may vary due to different number of TSCs and length of video.

3.5 Highlight Shot Similarity

Once the highlight shot is extracted from video, we can easily calculate the local feature vector for the highlight shot, $\vec{e_s} = (w_1, w_2, ..., w_K)$, where each w_i is the weight of each feature dimension for the highlight shot s. First, we calculate keyword count for the highlight shot, $n_s = (n_1, n_2, ..., n_{|\mathcal{W}|})$. Next, for each global emotional feature of video $e_{v,k} = (n_1, n_2, ..., n_{|\mathcal{W}|})$ with weight $w_{v,k}$, we calculate weighted cosine similarity for one feature dimension of highlight shot:

$$w_k = w_{v,k} \cdot cos(n_s, e_{v,i}), (1 \leq k \leq K) \qquad (4)$$

Algorithm 1 Highlight Shot Boundary Detection

```
for each frame $f_{v,i} \in v$ do
    calculate keyword strength vector $f_{v,i} = (g_1, g_2, ..., g_{|\mathcal{W}|})$
end for
for each global emotional feature $e_{v,j} \in e_v$ do
    for each frame $f_{v,i} \in v$ do
        calculate feature similarity $d_{f_{v,i}} = cos(f_{v,i}, e_{v,j})$
    end for
    find $max(d_{f_{v,k}})$ indexed by $k$
    if $d_{f_{v,k}} < \epsilon$ then
        continue
    end if
    set $n_l := k - 1$
    while $n_l \geq 1$ and $|d_{f_{v,n_l}} - d_{f_{v,n_l+1}}| < \delta$ do
        $n_l := n_l - 1$
    end while
    set $n_r := k + 1$
    while $n_r \leq N_v$ and $|d_{f_{v,n_r}} - d_{f_{v,n_r-1}}| < \delta$ do
        $n_r := n_r + 1$
    end while
    $\{f_{v,n_l}, ..., f_{v,n_r}\}$ is the highlight shot with feature $e_{v,j}$
end for
```

After that, for given two highlight shot, respectively with their local feature vector $\vec{e_{s_1}}$ and $\vec{e_{s_2}}$, we can measure their similarity easily and semantically through cosine similarity again.

4. EXPERIMENT AND EVALUATION

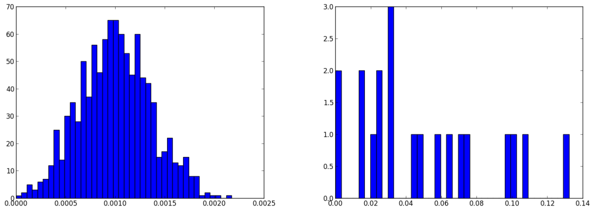
In the experiment, we propose a generative process for TSC data simulation so that we can generate simulated data including TSC and highlight shot. This is especially useful in our case, because TSC websites like bilibili do not provide highlight shots and we cannot evaluate our method by real data. Thus, we generate these simulation data that follows the characteristcs of real TSC data to test our method.

Algorithm 2 TSC Generative Process

```
1:  for all features $f \in [1, K]$ do
2:      sample feature mixture proportion $\vec{\varphi}_k \sim Dir(\vec{\beta})$
3:  end for
4:  for all videos $m \in [1, M]$ do
5:      sample video mixture proportion $\vec{\vartheta}_m \sim Dir(\vec{\alpha})$
6:      sample video length $L_m \sim Poisson(\lambda_1)$
7:      two-step highlights simulation (*)
8:      for all sequential timestamp $t$ of video $m$ do
9:          sample probability $p \sim uniform(0, 1)$
10:         if $t$ is in highlight and $p \in (0, \rho)$ then
11:             sample feature index $z_{m,t} \sim Mult(\vec{\vartheta}_m)$
12:             sample keyword $w_{m,t} \sim Mult(\vec{\varphi}_{z_{m,t}})$
13:         else if $t$ is not in highlight and $p \in (0, 1 - \rho)$ then
14:             sample feature index $z_{m,t} \sim Mult(\vec{\vartheta}_m)$
15:             sample keyword $w_{m,t} \sim Mult(\vec{\varphi}_{z_{m,t}})$
16:         end if
17:     end for
18: end for
```

As Algorithm 2 shows, lines 11-12 are the same as line 14-15, which is used to sample sentimental keywords at certain timestamp. The difference lies in the condition at line 10, dealing with the generation of useful comments in highlight areas and that at line 13, dealing with the generation of noise comments outside highlight shot. This is similar to real cases as video frames not in highlights is less attractive, users may be less likely to post comments. The possibility to post comment must depend on two reasons: a) to present a user's emotion triggered by previous stories or plots in the video; b) to randomly present his disaffect or boredom triggered by current periods of video. In the light of lower probability of generating comments in non-highlight areas than that in highlights and the same sampling process to get comments, two conditions can be merged into one and simplified into a single two-step sampling procedure. The result of this procedure merely influence the density of TSCs in different video frames, so it does not weaken the case of bag-of-words model to extract global features.

With this generative process, we first generate the sentimental dictionary with about 2000 unique keywords, which does not correspond to semantic sentimental meaning, but is represented by an identifier instead. As Figure 4a the probability of each term occurrence is sampled from the standard normal distribution and sums up to one. Next, we simulate 10 features that are randomly sampled from multinomial distribution. Each feature has a global probability to be selected for sampling referring to Figure 4b. The feature simulated here is only used for generating TSCs. Then, referring to the Algorithm 2, 40000 videos are simulated together with video length (mean=20), highlights amount (mean=4) and TSCs amount (mean=400). Notice that the highlights simulated here are used for training set that will be compared by estimated highlights. The simulation of TSCs is a little different because we want to add more noise data to test the effectiveness of our method. Useful TSCs in highlights are generated in the same way as the algorithm, whereas noise comments are uniformly sampled from dictionary on the whole video, which means they also occur in highlight areas.



(a) Keywords

(b) Features

Figure 4: Frequency of different probability

Then, we propose three evaluation indices to check whether the method is effective. As Figure 5 shows, coverage rate is used to evaluate coincidence degree between training highlights and estimated highlights. Our method provides a very high coverage rate up to 85.1% which is higher than the coverage rate 78.3% of frequency method. False positive rate is used to evaluate the accuracy of extraction. It represents the ratios of over-estimated highlights and non-highlight in original video. Feature Match Index is used to check whether the feature of estimated highlight matches the given feature of the significant feature of query highlight. One problem here is that basic LDA model for training global feature is an unsupervised learning method, which leads to the result that the simulated features are unable to correspond to the estimated features one by one. Thus, one simple alternative is to traverse all estimated features in finding possible feature of highlights.

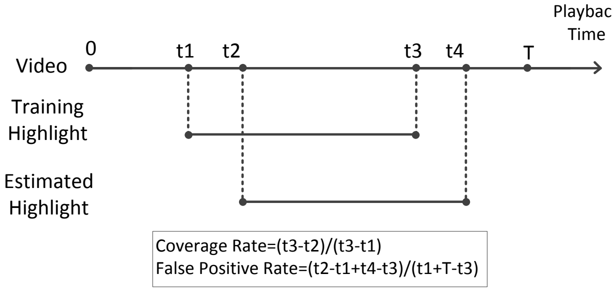

Playback Time
Video
0
t1
t2
t3
t4
T
Training Highlight
Estimated Highlight
Coverage Rate=(t3-t2)/(t3-t1)
False Positive Rate=(t2-t1+t4-t3)/(t1+T-t3)

Figure 5: Coverage Rate Example

5. CONCLUSION AND FUTURE WORK

In conclusion, we propose a new method for video highglight shot extraction and retrieval using TSCs. TSC is featured by short-text, real-time and subjectivity which can implicitly reflect user feeling on a video. To the best of our knowledge, this is the first time that such textual information was utilized on video extraction. In our method, we first extract latent global sentimental features from TSCs through classical LDA model. With these features, we then propose the Central Diffusion algorithm to detect highlight boundaries. The result of simulation experiment shows that the matched rate is pretty high. This proves that our method is quite effective in solving such problem.

In future work, these aspects can be improved. First, a timestamp can also be regarded as a variable and basic LDA model can be adapted to fit the situation. The difficulty lies in the parameter inference and whether the result is better than this naive approach. Second, each TSC can be represented as multiple words just like comments real world. Third, since we largely rely on users emotion, there is the possibility that users will post TSC arbitrarily or with latency. Experiment with real data is required to be conducted to evaluate the method.

6. ACKNOWLEDGEMENTS

This work was supported by the Youth Science and Technology Foundation of Shanghai (NO. 15YF1412600).

7. REFERENCES

[1] Y. Aytar, M. Shah, and J. Luo. Utilizing semantic word similarity measures for video retrieval. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[3] Y. Y. Chung, W. J. Chin, X. Chen, D. Y. Shi, E. Choi, and F. Chen. Content-based video retrieval system using wavelet transform. *WSEAS Transactions on Circuits and Systems*, 6(2):259–265, 2007.

[4] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[5] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs. *ICWSM*, 7, 2007.

[6] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

[7] M. Hamasaki, H. Takeda, T. Hope, and T. Nishimura. Network analysis of an emergent massively collaborative creation community. In *Proceedings of the Third International ICWSM Conference*, pages 222–225, 2009.

[8] G. Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[9] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank. A survey on visual content-based video indexing and retrieval. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 41(6):797–819, 2011.

[10] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[11] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[12] A. F. Smeaton. Techniques used and open challenges to the analysis, indexing and retrieval of digital video. *Information Systems*, 32(4):545–559, 2007.

[13] B. Wu, E. Zhong, B. Tan, A. Horner, and Q. Yang. Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 721–730. ACM, 2014.

[14] K. Yoshii and M. Goto. Musiccommentator: Generating comments synchronized with musical audio signals by a joint probabilistic model of acoustic and textual features. In *Entertainment Computing–ICEC 2009*, pages 85–97. Springer, 2009.