Crowdsourced Time-sync Video Tagging using Temporal and Personalized Topic Modeling

Bin Wu[1], Erheng Zhong[1], Ben Tan[1], Andrew Horner[1], Qiang Yang[1,2]
[1]Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Hong Kong
[2]Noah's Ark Lab, Huawei, Hong Kong
{bwuaa,ezhong,btan,horner,qyang}@cse.ust.hk

ABSTRACT

Time-sync video tagging aims to automatically generate tags for each video shot. It can improve the user's experience in previewing a video's timeline structure compared to traditional schemes that tag an entire video clip. In this paper, we propose a new application which extracts time-sync video tags by automatically exploiting crowdsourced comments from video websites such as Nico Nico Douga, where videos are commented on by online crowd users in a time-sync manner. The challenge of the proposed application is that users with bias interact with one another frequently and bring noise into the data, while the comments are too sparse to compensate for the noise. Previous techniques are unable to handle this task well as they consider video semantics independently, which may overfit the sparse comments in each shot and thus fail to provide accurate modeling. To resolve these issues, we propose a novel temporal and personalized topic model that jointly considers temporal dependencies between video semantics, users' interaction in commenting, and users' preferences as prior knowledge. Our proposed model shares knowledge across video shots via users to enrich the *short* comments, and peels off user interaction and user bias to solve the *noisy*-comment problem. Log-likelihood analyses and user studies on large datasets show that the proposed model outperforms several state-of-the-art baselines in video tagging quality. Case studies also demonstrate our model's capability of extracting tags from the crowdsourced *short* and *noisy* comments.

Categories and Subject Descriptors

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

Keywords

Video tagging; crowdsourcing; topic modeling; temporal and personalized model

1. INTRODUCTION

Online videos have become indispensable to peoples' daily lives. Everyday, millions of people watch online videos for entertainment, news, and education. Traffic created by online video websites such as Youtube, Hulu, and Netflix occupied 56.6% of the total global consumer internet traffic in 2012 [1]. At the same time, the volume of online videos is extremely large. On YouTube, over 6 billion hours of video are watched each month and 100 hours of video are uploaded to YouTube every minute [2]. The huge traffic and volume of online videos have made data management and indexing, the key parts of video searching, very challenging.

Figure 1: Videos with time-sync descriptions improve the user's experience in previewing and locating video content. We magnified the thumbnails and time-sync description for illustration.

To solve the aforementioned problems, automatic video tagging techniques have been proposed to generate keywords to represent a video for fast and accurate video indexing [20, 17]. However, these techniques can only provide video-level tags, that is, keywords corresponding to entire video clips. The problem is that even if the generated tags can perfectly summarize the video content, users have no idea how these tags are associated with the video playback time, which results in a long wait for video buffering, and having to either slide through the entire video or randomly approximate the informativeness of the video content.

To this end, time-sync video tagging has been proposed as a new paradigm. Time-sync video tags are synchronized to a video's playback time, and are therefore well structured in a timeline manner. Users will be able to better preview and search videos. For example, Youku.com, one of the biggest online video websites in China, has started to provide such a feature on some videos. As shown in Figure 1[1], users can preview video content in both thumbnails and text/tags by indicating the playback time. Also, this textual information can enrich search results with playback time positions.

[1]http://goo.gl/FoI6t, accessed 9 July.

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623625.

In addition to improving user experience and indexing precision, time-sync tags and the corresponding video shots also act as an accurate labeled set for extrinsic tasks such as video classification. All in all, time-sync video tagging offers several advantages compared to traditional video tagging schemes which only tag for the entire video.

However, generating time-sync video tags automatically usually requires image to text transformation [19] and subtitles/annotations, both of which are either too difficult to realize or too expensive to obtain. Fortunately, video sharing websites such as Nico Nico Douga[2] and acfuntv[3], where users can comment on playback timestamps, provide opportunities to solve this problem. In such video websites, comments are overlaid directly over the video, synchronized to a specific playback time. This allows comments to respond directly to the corresponding video semantics, in sync with users - creating a sense of shared watching experience. For simplicity, we call this type of online video as time-sync commented (TSC) videos. In this paper, our objective is to extract time-sync video tags for TSC videos using comments only, which is a new application.

Given the well-structured comments of TSC videos, the learning problem of time-sync video tagging can be regarded as topic extraction for each video shot. Nevertheless, extracting tags from TSC videos is not only a new application but also brings up a new crowdsourcing problem. The challenge in TSC videos is that crowd users with different preferences interact with one another frequently and bring noise into the data, while the comments are too sparse to compensate for the noise, i.e., the text content in each video shot is very *short* and *noisy*. More specifically, there are only ten comments for each video shot on average, and each comment normally contains less than five words. This is because in TSC videos, each comment is only allowed to stay on the screen for a few seconds, which restricts users to *short* comments. Comments are also *noisy* because they usually contain information from multiple sources. As in traditional crowdsourcing problems [22, 23], users may have their own preferences on topics, which are not necessarily related to the current video semantics, and therefore introduce user bias in their comments. Moreover, the shared video watching experience allows users to interact with one another. In many cases, users write irrelevant comments such as replying to a previous comment, which may bury the most valuable information.

Previous methods cannot be applied to solve the *short* and *noisy* issues. Some researchers have taken advantage of the huge number of real-time comments for big events generated by crowds on social media applications such as Twitter, which are similar to TSC videos due to their *short* and *noisy* properties. For instance, Chakrabarti *et al.* proposed to summarize key tweets for live football video events [5]. However, the videos they explored were typically live videos such as big sports games which are important enough for large number of related tweets in real-time. Extending these schemes to online video tagging is more difficult because of the lack of viewers and comments. More importantly, since these approaches did not consider knowledge enrichment and user modeling in mining crowdsourced content, the *short* and *noisy* (e.g., user bias and interaction) issues cannot be solved.

Some unsupervised methods have been proposed to consider knowledge sharing across instances in crowdsourced data and collecting high quality labels by integrating noisy-labels. They achieved this goal by modeling information such as labeling ability [22, 23], by assuming that users' labeling follows an Independent Identical Distribution (*i.i.d.*). Nevertheless, in a TSC video, users can see previous comments before commenting, which means labeling is not independent (commenting can be regarded as labeling on video semantics). In fact, the dependencies between users' labels might have strong effects on modeling users' preferences. Intuitively, a user might have a preference on a specific topic if s/he often comments about it, while users who follow this topic may be simply responding to the previous comments, not necessarily indicating the same preference. Ignoring the interactions in comments may lead to inaccurate modeling of users' preferences, and fail to remove user bias in topic extraction. Therefore, the typical *i.i.d.* assumption does not hold for users' labeling, and thus previous methods may not be suitable in TSC videos.

To solve these problems, we propose an unsupervised method to automatically generate time-sync video tags using crowdsourced comment data only. More specifically, the technical contribution of this paper is that we build a novel temporal and personalized topic model which integrates users' preferences, users' interaction, and the temporal semantics of videos. On one hand, it encodes the temporal semantics correlation between successive video shots, which can enrich the *short* comments of the current video shot. On the other hand, to recover topics of the current video semantics accurately, our proposed model removes user interaction and user bias to address the *noisy* problem. This is achieved by encoding semantics dependencies between comments within the same shot, and utilizing an adaptive variable of each user to denote their latent preferences that decide their global commenting preferences.

The main contributions of our paper are as follows:

1. We propose a novel time-sync video tagging application for time-sync commented videos. To the best of our knowledge, this is the first work on automatic time-sync video tagging using video comments only.
2. We propose a novel temporal and personalized topic model for automatic video tagging, which addresses *short* and *noisy* (i.e., user bias and interaction) problems of time-sync comments.
3. We evaluate our proposed model with real-world large datasets, user studies, and a case study. The results show that our proposed model outperforms baselines in terms of tagging quality with similar computational complexity.

2. PROBLEM DEFINITION

In this section, we first present an example video to illustrate what is a TSC video. Then, we define our problem formally. After the definition, we show statistics on TSC videos to give some insights on the data.

2.1 Illustration of Time-sync Commented Videos

Two snapshots (with a one-minute gap) of an example video[4] are shown in Figure 2. Users can write comments with respect to the current video semantics (e.g., 'bento', 'shrimps'). Moreover, users' views can be affected by previous comments. For example, in the first snapshot, user B's comment "...SHIMPS..." may help other users (e.g., user A) recognize the unobvious shrimps. Then, the comment "Eating the shrimps" provided by user D is probably generated under the co-effect of both previous comments ("shimps") and the current semantics ("eating"). Note that users' IDs can be retrieved by parsing the website. In this paper, we refer to such videos as time-sync commented (TSC) videos.

[2]http://www.nicovideo.jp/. Nico Nico Douga is the most popular video sharing website in Japan, with 25 million users (19.6% of Japnese population) according to the statistics in May, 2012.

[3]http://www.acfun.tv

[4]http://live.nicovideo.jp/watch/lv139636921

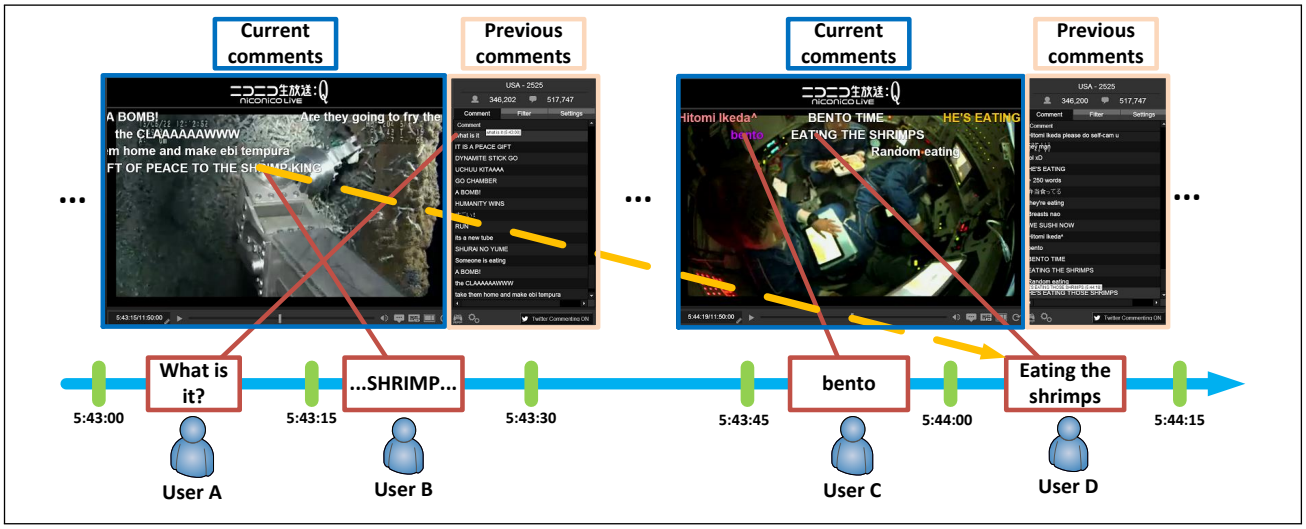


Figure 2: Snapshot of an example Time-Sync Commented (TSC) video. Users share their watching experience by providing time-sync comments that appear on the screen.

Table 1: Notation List

Notation	Definition
V	Set of videos, containing $\|V\|$ videos
U	Set of users, containing $\|U\|$ users
v	A video, containing $\|v\|$ shots
s	A shot, containing $\|s\|$ comments
v_s	Video which contains s
c	A comment, containing $\|c\|$ words
w	A word
W	Vocabulary set
u_c	User who commented comment c
s_c	Shot which contains comment c
pre_s	Preceding shots of s in v_s
pre_c	Preceding comments of c in s_c
T	Number of tags to be generated for each shot
m_{pre_s}	Semantics prior distribution of s, generated by pre_s
m_{pre_c}	Semantics prior distribution of c, generated by pre_c
K	Number of hidden topics
N	Number of comments
M	Size of vocabulary
$\boldsymbol{x_u}$	u's preference vector.
$\boldsymbol{\lambda_s}$	s's topic prior distribution vector.
$\boldsymbol{\alpha_c}$	Dirichlet prior distribution vector for c's topic distribution.
β	Dirichlet prior for word-topic distribution.
$\boldsymbol{\varphi_t}$	Word distribution of topic t
$\boldsymbol{\pi_c}$	c's topic prior distribution base vector
$\boldsymbol{\theta_c}$	c's topic distribution

2.2 Formal Problem Definition

In this section, we present the problem definition formally. We have a set of videos $V = \{v_1, v_2, \ldots, v_{|V|}\}$ of size $|V|$, and users $U = \{u_1, u_2, \ldots, u_{|U|}\}$ of size $|U|$ who have written comments on these videos V. Each video $v \in V$ has a number of shots which have been segmented previously, $v = \{s_{(v)1}, s_{(v)2}, \ldots, s_{(v)|v|}\}$ where $|v|$ denotes the number of shots in video v. Shots in a video are organized according to playback time. For example, $s_{(v)i}$ means the i-th shot appears in video v and $s_{(v)i+1}$ is the $(i+1)$-th shot. We denote a set of shots s as $pre_s = \{s'|s, s' \in v, s'$ appears before $s\}$. Each shot has a number of comments $s = \{c_{(s)1}, c_{(s)2}, \ldots, c_{(s)|s|}\}$, where $|s|$ is the number of comments in shot s. Note that each comment corresponds to a specific timestamp in a video, and all comments are organized according to playback time. Users may write their comments after seeing the preceding comments, and the set of preceding comments is denoted as $pre_c = \{c'|c, c' \in s, c'$ appears before $c\}$. Each comment c consists of a set of words $c = \{w_{(c)1}, w_{(c)2}, \ldots, w_{(c)|c|}\}$, where $w_{(c)i}$ is the i-th obseved word in c and $|c|$ denotes the number of words in comment c. Usually, both $|s|$ and $|c|$ are small. For each comment c, the user is known and denoted as u_c. Similarly, we denote the shot containing comment c as s_c. We also denote the total number of comments as N and the size of the vocabulary as M for simplicity.

Given that comments in each video shot are *short* and *noisy* (i.e., $|s|$ is small and each c is affected by pre_c), our task is to extract tags $W_s = \{w_1, w_2, \ldots, w_T\}$ from each shot s so that the tags well describe the shot, that is to find:

$$W_s = \max_{W'_s \subset W} P(W'_s|s, u_c, Pre_c, Pre_s), \forall c \in s. \quad (1)$$

where W is the vocabulary set, and W'_s is a candidate of W_s. For reference, the above notations have been listed in Table 1.

2.3 Inspection of Time-Sync Commented Videos

For better insight into the time-sync comments, we have conducted data inspection of the video comments from acfuntv[5], a typical time-sync commented video website. Statistics of pairwise Jaccard similarity [14] and the number of Chinese characters in the comments have been investigated, as shown in Figure 3 and Table 2.

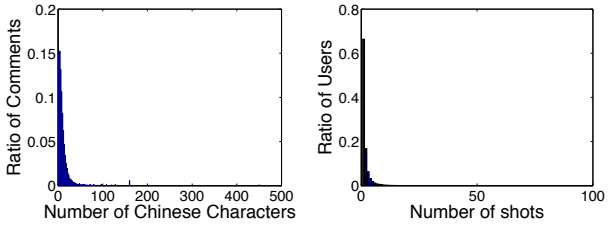
Table 2: Jaccard Similarity for the Average Comment

Statistics	Average Comments Jaccard Similarity
Overall Comments	0.008
Intra-user	0.036
Overall Shots	0.038
Intra-shot	0.065
Comment with Preceding Comments	0.080
Shot with Preceding Shots	0.099

The number of Chinese characters is usually small in most comments, i.e., about 15 characters on average (see Figure 3(a)). The average similarity between a comment and its preceding comments is much higher than the intra-shot similarity (the average pairwise

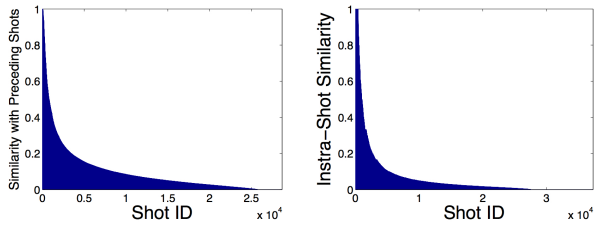
[5]http://www.acfun.tv

similarity between comments within the same shot), which indicates the tight correlation between a comment and its preceding comments. Moreover, a large number of comments are very similar, e.g., with an average similarity higher than 0.8 compared to their preceding comments (see Figure 3(f)). These two observations numerically describe the *short* comment property and user-interactions for time-sync comments.



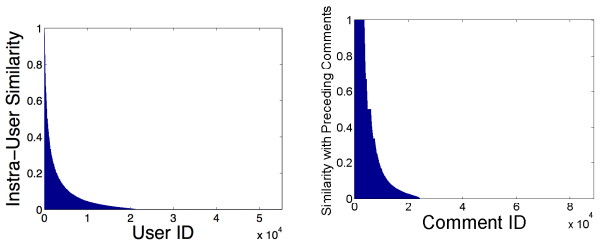
(a) Number of characters in each comment.

(b) Number of shots for each user.



(c) Similarity between a shot and its preceding shot(s).

(d) Similarity between comments for each shot.



(e) Similarity between comments for each user.

(f) Similarity between a comment and its preceding comment(s).

Figure 3: Data Inspection of Time-Sync Comments.

On the other hand, as shown in Table 2, the average intra-user (i.e., comments by the same user) similarity (0.036) and average inter-shot similarity (0.099) are much higher than the overall similarity (0.008). This observation indicates users' preferences in comments and the temporal dependency of video semantics.

3. EXTRACTING TAGS FROM TIME-SYNC COMMENTED VIDEOS

We propose to exploit knowledge from the preceding video shots to solve the *short* and *noisy* comment problems. On one hand, we enrich the knowledge in the current video shot by considering its temporal dependencies. On the other hand, we model user preferences and interactions to remove content-irrelevant data from each comment. Specifically, we consider the generative process of a comment as a probabilistic model, where the words in comments are observed and the underlying video semantic topics are hidden. To better infer the hidden topics, we incorporate the above factors as prior knowledge in generating topics, such that knowledge can be shared across videos and shots accurately through users.

Topic modeling, which aims to extract semantically valid topics from document collections, is a natural choice for solving the problem. Before describing our proposed model, we first briefly introduce Latent Dirichlet Allocation [4], the most well-known topic model which has been successfully applied in text analysis tasks such as tag recommendation [13].

3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) assumes words in a document/comment (for simplification, hereafter referred to as comment) are generated by some hidden topics. For example, a comment containing "soccer", "NBA", "Kobe Bryant" is probably about sports while "predicting" and "LDA" are probably about machine learning. LDA aims to infer hidden topics of a given comment. For instance, what topic or mixture of topics would "soccer" and "predicting" be about? Are they about sports, machine learning, or both? More specifically, LDA represents the mixture of topics as a probability distribution over topics. Note that LDA is an unsupervised method, and the topics are latent variables.

3.2 Temporal and Personalized Topic Model

In this section, we introduce our proposed temporal and personalized topic model (TPTM). Consider when user u writes comment c on video shot s. On the one hand, both user u's preference and video shot s's semantics determine the prior knowledge of comment c's topics, making the process personal. On the other hand, since user u can see the preceding comments in s_c when generating comment c, c is also affected by its preceding comments. Moreover, preceding shots are semantically similar to current shots with high probability. The temporal dependencies between comments and the similarity between shots' semantics make the process temporal.

Formally, we denote $\mathcal{MN}()$ as a sampling process with respect to Multinomial distribution, $\mathcal{D}(\alpha)$ and $\mathcal{D}(\beta)$ as sampling processes with respect to a Dirichlet distribution with parameter α for per-comment topic distributions and β for per-topic word distributions. The generative process of TPTM is as follows, and the corresponding graphical model is shown in Figure 4:

1. For each user u
   (a) Generate u's **preference vector** $\boldsymbol{x_u} \sim \mathcal{N}(\mathbf{0}, \sigma_u \boldsymbol{I_K}), \boldsymbol{x_u} \in \mathbb{R}^K$
2. For each video v
   (a) For each shot $s \in v$
      i. Generate s's **topic prior distribution vector** $\boldsymbol{\lambda_s} \sim \mathcal{N}(\boldsymbol{m_{pre_s}}, \sigma_s \boldsymbol{I_K}), \boldsymbol{\lambda_s} \in \mathbb{R}^K$
3. For each topic t
   (a) Generate the word distribution for topic t $\boldsymbol{\varphi_t} \sim \mathcal{D}(\beta)$
4. For each comment $c \in s$, where c is commented by u_c
   (a) Let c's **topic prior distribution base vector** be $\boldsymbol{\pi_c} = \boldsymbol{\lambda_{s_c}} \odot \boldsymbol{x_{u_c}} + \boldsymbol{m_{pre_c}}$
   (b) Let c's **topic prior distribution vector** be $\boldsymbol{\alpha_c} = lgt(\boldsymbol{\pi_c})$
   (c) Generate c's topic distribution $\boldsymbol{\theta_c} \sim \mathcal{D}(\boldsymbol{\alpha_c})$
   (d) For each observed token $w_j \in c$
      i. Generate $z_{cj} \sim \mathcal{MN}(\boldsymbol{\theta_c})$

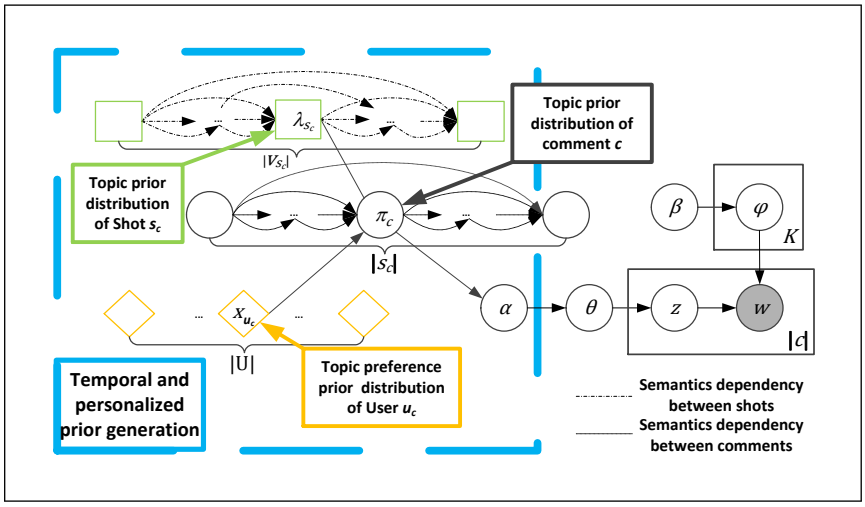


Figure 4: The graphical model of our temporal and personalized topic model (TPTM). For simplification, we only show a simple example where a user u_c writes a comment c on a shot s_c. In the exact model, a shot can contain multiple comments while a user can write multiple comments on multiple shots and videos.

ii. Generate $w_j \sim \mathcal{MN}(\boldsymbol{\varphi}_{\boldsymbol{z_{cj}}})$

where $\odot$ denotes element-wise multiplication, and $lgt(y) = log[1 + e^y]$. $\boldsymbol{m_{pre_s}}$ is defined as the average temporal semantics of pre_s, the preceding shots of shot s, acting as prior distribution of s's semantics. As shots that are closer in sequential order tend to be more similar in semantics, we assume a semantic similarity between shot s and shots in pre_s subject to an exponential decay, which is formally defined as:

$$\boldsymbol{m_{pre_s}} = \frac{\sum_{s' \in pre_s} \exp^{-\gamma_s \Delta(s,s')} \boldsymbol{\lambda_{s'}}}{\sum_{s' \in pre_s} \exp^{-\gamma_s \Delta(s,s')}} \tag{2}$$

where $\Delta(s, s')$ is the absolute difference in appearance order between s and one of its preceding shots s', γ_s is the decay rate. Similarly, $\boldsymbol{m_{pre_c}}$ is defined as:

$$\boldsymbol{m_{pre_c}} = \frac{\sum_{c' \in pre_c} \exp^{-\gamma_c \Delta(c,c')} \boldsymbol{\pi_{c'}}}{\sum_{c' \in pre_c} \exp^{-\gamma_c \Delta(c,c')}} \tag{3}$$

We observed that both $\boldsymbol{m_{pre_s}}$ and $\boldsymbol{m_{pre_c}}$ are essential for dealing with the *short* and *noisy* comment challenges. $\boldsymbol{m_{pre_c}}$ explicitly models the user interaction by encoding semantic dependencies between comments within the same shot, which peels off co-user interference and makes the extracted video topics less *noisy*. Also, on the one hand, videos are connected via the common users, where knowledge can be implicitly propagated via modeling users' latent preference x_u adaptively; on the other hand, modeling temporal dependency between shots (i.e., $\boldsymbol{m_{pre_s}}$) encodes semantics gained from preceding shots. Such knowledge propagation enriches the current shot semantics and therefore addresses the *short* comment problem.

3.3 Inference

The inference of TPTM has two steps: 1) infer the topic distribution for each comment; 2) extract the most probable words for each shot. The first step can be achieved by maximizing the joint distribution $P(z, \lambda, x, \theta, \varphi)$. In the second step, to extract the most probable words of a shot s, we generate topics from the topic distribution $\mathcal{T}_s$ of s obtained by averaging each comment c's topic distribution θ_c, where $c \in s$. Then, tags of shot s are extracted by picking the most probable words.

Observe that for each comment c, if α_c is fixed (i.e., π_c is fixed), the model is equivalent to multiple independent LDAs, where each comment has a different prior distribution for each topic. Therefore, we can perform a collapsed Gibbs sampling by integrating out θ and φ, and sampling z. After integrating out θ and φ, the complete likelihood $P(z, \lambda, x)$ is given by:

$$\begin{aligned} P(z,\lambda,x) =& P(z|\lambda,x)P(\lambda)P(x) \\ =& \prod_{c:c\in s_c, s_c\in v, v\in V} \left\{ \frac{\Gamma\left(\sum_t lgt(x_{u_ct}\lambda_{s_ct} + m_{pre_{ct}})\right)}{\Gamma\left(\sum_t lgt(x_{u_ct}\lambda_{s_ct} + m_{pre_{ct}}) + n_c\right)} \right. \\ & \prod_t \frac{\Gamma\left(lgt(x_{u_ct}\lambda_{s_ct} + m_{pre_{ct}}) + n_{t|c}\right)}{\Gamma\left(lgt(x_{u_ct}\lambda_{s_ct} + m_{pre_{ct}})\right)} \\ & \prod_t \frac{1}{\sqrt{2\pi\sigma_s}} \exp\left(-\frac{(\lambda_{s_ct} + m_{pre_{s_ct}})^2}{2\sigma_s^2}\right) \\ & \left. \prod_t \frac{1}{\sqrt{2\pi\sigma_u}} \exp\left(-\frac{x_{u_ct}^2}{2\sigma_u^2}\right) \right\} \end{aligned} \tag{4}$$

Here we maximize $P(z, \lambda, x)$ with respect to z, λ, and x respectively. A Gibbs sampler as described in [11] is used to sample z, and λ and x are updated using gradient descent. As for λ, the derivative of the log of equation Equation (4) with respect to λ_{st} given shot s and topic t is:

$$\begin{aligned} \frac{\partial P(z,\lambda,x)}{\partial \lambda_{st}} =& -\frac{\lambda_{st} + m_{pre_{st}}}{\sigma_s^2} \\ & + \sum_{c:s_c=s} \left\{ x_{u_ct} dlgt(x_{u_ct}\lambda_{st} + m_{pre_{ct}}) \right. \\ & \times \left[\Psi\left(\sum_t lgt(x_{u_ct}\lambda_{st} + m_{pre_{ct}})\right) \right. \\ & - \Psi\left(\sum_t lgt(x_{u_ct}\lambda_{st} + m_{pre_{ct}}) + n_c\right) \\ & + \Psi\left(lgt(x_{u_ct}\lambda_{st} + m_{pre_{ct}}) + n_{t|c}\right) \\ & \left. \left. - \Psi\left(lgt(x_{u_ct}\lambda_{st} + m_{pre_{ct}})\right) \right] \right\} \end{aligned} \tag{5}$$

where $dlgt(y) := \partial lgt(y)$. Similarly, the derivative with respect to x_{ut} given user u and topic t is:

$$\frac{\partial P(z,\lambda,x)}{\partial x_{ut}} = -\frac{x_{ut}}{\sigma_u^2} + \sum_{c:u_c=u}\Big\{\lambda_{s_ct} dlgt(x_{ut}\lambda_{s_ct} + m_{pre_{ct}}) \times \Big[\Psi\big(\sum_t lgt(x_{ut}\lambda_{s_ct} + m_{pre_{ct}})\big) - \Psi\big(\sum_t lgt(x_{ut}\lambda_{s_ct} + m_{pre_{ct}}) + n_c\big) + \Psi\big(lgt(x_{ut}\lambda_{s_ct} + m_{pre_{ct}}) + n_{t|c}\big) - \Psi\big(lgt(x_{ut}\lambda_{s_ct} + m_{pre_{ct}})\big)\Big]\Big\} \quad (6)$$

where $m_{pre_{ct}}$ can be regarded as a constant given user u_c since pre_c does not contain comments written by u_c (according to our definition). Finally, the updating equations for λ_{st} and x_{ut} are:

$$\lambda_{st} \leftarrow \lambda_{st} - \eta\frac{\partial P(z,\lambda,x)}{\partial \lambda_{st}} \quad (7)$$

$$x_{ut} \leftarrow x_{ut} - \eta\frac{\partial P(z,\lambda,x)}{\partial x_{ut}} \quad (8)$$

where η is the learning rate. The inference procedure, inspired by [15], is as follows: fixing λ and x, the model is equivalent to multiple independent LDAs where the standard LDA Gibbs sampler is used to sample z and φ. After several iterations of sampling, we alternatively update λ and x using Eq (7) and Eq (8). The implementation details is described in Section 4.2.

After inference, we can obtain the topic distribution $\mathcal{T}_{st}$ of each shot s given a topic t:

$$\mathcal{T}_{st} = \frac{\sum_{c\in s}\theta_{ct}}{|s|} \quad (9)$$

Then, the T most probable words can be extracted from shot s. The probability of choosing a word is defined as:

$$P(w|s) = \sum_t \mathcal{T}_{st} * \varphi_{tw} \quad (10)$$

Framework. The complete algorithm of our proposed temporal and personalized model is shown in Algorithm 1. Overall, it is an iterative process. In each iteration, Gibbs sampling is used to infer lower level variables such as the topic distribution θ, word distribution for topic φ, and topic z, which is identical to LDA. For every 200 iterations, higher level variables such as user preference x and video shot semantics prior distribution λ are updated in turn with respect to the joint distribution $P(z,\lambda,x)$ using Eq. (7) and Eq. (8). Then the temporal semantics prior distributions m_{pre_s} and m_{pre_c} are updated.

Time complexity. The time complexity of Gibbs sampling for lower level variables is $\mathcal{O}(NMK)$ for each iteration. The complexity of updating λ and x for each iteration are both equal to $\mathcal{O}(NK)$. For updating m_{pre_s} and m_{pre_c}, the upper bounds are $\mathcal{O}(N^2K)$ and $\mathcal{O}(\sum_{v\in V}|v|^2K)$, respectively. Actually, these bounds can be much tighter in our problem setting. Since the average number of comments $|s|$ for each shot s and the average number of shots $|v|$ for each video v is about 10 (see Table 3), and both m_{pre_c} and m_{pre_s} are close to 0 when $\Delta \approx 10$ (according to Eq. (2) and Eq. (3)), the complexity of updating m_{pre_c} and m_{pre_s} can be further bounded as $\mathcal{O}(10NK) < \mathcal{O}(NMK)$. Therefore, the complexity of TPTM is $\mathcal{O}(NMK)$.

Algorithm 1 Temporal and Personalized Topic Model

Input Videos V; each video $v \in V$ contains shots; each shot $s \in v$ contains comments; each comment c corresponds to a specific playback timestamp, with observed words and a known user $u_c \in U$. Number of topics K.
Output Time-sync tags of each shot s.
Given a topic t, initialize the topic prior distribution of shot $\lambda_{st} \sim \mathcal{N}(0,\sigma_s^2)$, initialize the user preference vector $x_{ut} \sim \mathcal{N}(0,\sigma_u^2)$.
for i = 1 to #sampling iterations **do**
  if i is the odd multiples of 200 **then**
    Update λ_{st} using Eq. (7).
  else if i is the even multiples of 200 **then**
    Update x_{ut} using Eq. (8).
  end if
  if i is the integer multiples of 200 **then**
    Update m_{pre_s} and m_{pre_c} using Eq. (2) and Eq. (3).
  end if
  Sample topic of each given observed token in c.
end for
for each shot s **do**
  Extract time-sync tags using Eq. (10).
end for

4. EXPERIMENTS

We empirically answer the following questions in this section: 1) Does the proposed temporal and personalized model generate better tags for video shots? 2) How do the model parameters (e.g., the number of topics K and the decay rate γ) affect the model performance? To answer these questions, we first introduce the data we used. We then compare the performance of our method to some state-of-the-art methods. We use log-likelihood as the evaluation metric for both 1) and 2), which is described in detail in Section 4.3. Moreover, we conduct human evaluation and a case study to show the quality of the tags generated by our method.

4.1 Data Description

Our data was retrieved from a Chinese TSC video website[6]. We use two datasets for experimental studies: comments for videos uploaded in the music section[7] and the fun section[8] snapshot on Oct 2012, with 9,992 videos and 10,187 videos, respectively. Tokenizing and stemming of the raw comments were done by a Chinese natural language processing toolbox, ICTCLAS[9]. However, since comments in TSC videos contain a large amount of internet slang, the resulting tokens missed many meaningful words, and had a large number of single Chinese characters due to incorrect tokenization. Although some single characters have their own merits, most had negative impact according to our experiments. Therefore, to best recover the internet slang, we applied a bigram concatenation of the previous results and obtained a new vocabulary. All single characters were then deleted and about 300,000 words remained. Next, we segmented each video according to the number of comments over playback time. We used a peakfinder[10] to find the peaks and troughs of the comments density, and then segment the videos into shots. Each shot contained at least eight comments, and each video contained at least one shot. Videos not satisfying these requirements were filtered out[11]. Finally, there were 6922 videos, 46,078 shots, 420,125 comments, and 150,838 users for music videos; 9492 videos, 68,069 shots, 683,759 comments, and

[6] http://www.acfun.tv/
[7] http://www.acfun.tv/v/list58/index.htm
[8] http://www.acfun.tv/v/list60/index.htm
[9] http://www.ictclas.org/index.html
[10] http://www.mathworks.com/matlabcentral/fileexchange/25500-peakfinder
[11] For videos without comments where our approach (text-based) is not applicable, content-based approaches such as [9] can be adopted.

Table 3: Summary of data.

Type	Number	Average Number
Data: Music Videos		
Videos	6922	-
Users	150,838	-
Shots	46,078	6.7 per video
Comments	420,125	9.11 per shot
Data: Fun Videos		
Videos	9492	-
Users	231,914	-
Shots	68,069	7.3 per video
Comments	683,759	10.04 per shot

231,914 users for fun videos[12]. 14% of the users published more than 3 comments, and these users occupied 60% of all comments. This means that there are many long-tail users, which makes the problem even more challenging. The data is summarized in Table 3.

4.2 Experimental Setup

For comparison purposes, we introduce two baselines: LDA and Sembler [23]. Sembler has been proposed to integrate crowd-sourced labels for tasks with temporal content such as name entity recognition (NER) and part of speech tagging (POS). For implementation, we mimic Sembler by modifying TPTM to ignore the dependencies between users. Note that Algorithm 2 described in [23] was not adopted because the labeling space in our problem setting is exponential to the vocabulary size, which makes it computationally intractable to generate valid sequential labelings. Also, we did not introduce Dynamic Topic Models (DTM) as baselines due to two reasons: 1). DTM models the dynamics of the prior β, which is not applicable for our problem setting as each video shot can be commented at different time stamps; 2). DTM is also a simplified version of Sembler in modeling α by ignoring the user preference $\boldsymbol{x_u}$.

The model parameters of each baseline are described as follows: In LDA, each comment was treated as an independent document, and the prior parameter α was fixed at 0.5. In Sembler, videos are considered temporal while users' labelings are assumed to follow an independent identical distribution (*i.i.d.*). That is, comments are assumed to be independent of preceding comments. More specifically, we set $m_{pre_{ct}} = 0$ for each comment c and topic t. In TPTM, dependencies between shots and between users' comments are modeled. We implemented these three models based on a Gibbs Sampler, with the number of sampling iterations set to 1000, α_c for each comment initialized to 0.5, and β to 0.1. In TPTM, we alternatively optimized λ and x using Eq. (7) and Eq. (8) (see Algorithm 1). In Sembler, only λ is optimized five times. The learning rate in both TPTM and Sembler is defined as $\eta = \frac{0.1}{2^{i\%10}}$, where i is the current iteration number. γ_s and γ_c were both set to 1.

We observe that the computational time increases linearly with larger data size for all three methods. For TPTM on full music video data, every 200 iterations of Gibbs sampling and one iteration of optimizing λ or x take about 1.5 hours in our computer, which has 16G of memory and a 3.2 Gz CPU, while LDA takes about 1 hour. All three methods converge within 600 Gibbs sampling iterations.

Table 4 shows the top words for four music video topics[13]. The first topic is about the background music of videos; the second describes the video structure[14]; the third topic refers to comments on music videos concepts; the fourth is about the well-known song 'Gangnam Style'.

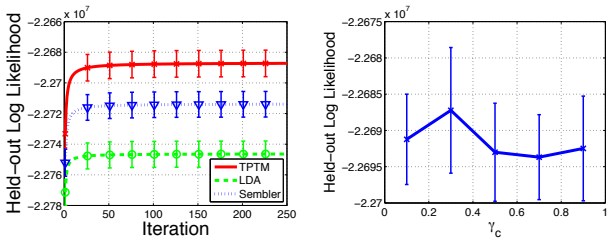
[12]We have made the data publicly available at http://www.cse.ust.hk/~bwuaa/TSC/TSC.zip.

[13]These words were manually translated to English by the authors.

4.3 Performance Analyses

We examined the held-out log-likelihood $P(\mathbf{c}'|\alpha, \varphi)$ of the three methods, where $\mathbf{c}'$ is the held-out comment set, and α and φ are obtained in the training process. The higher the held-out log-likelihood, the better the model predicts the topic distribution of the held-out data [21]. We used the first half videos for the training set and the second half for the held-out set. More specifically, if a video was commented on only by new users (i.e., users who did not write comments in the training set), no prior knowledge of personal information was available for these new users. Therefore, we only considered the 108,364 users who provided comments in both the training and held-out sets. The results are a ten run average.

As shown in Figure 5(a), both TPTM and Sembler achieved higher held-out log-likelihood as compared to LDA, which shows the benefits of modeling user preferences and the temporal dependencies between video shots. Moreover, on average, the initial log-likelihoods of TPTM and Sembler are already notably better than or close to the converged log-likelihood of LDA, which indicates the accurate estimation of α by the first two methods. Furthermore, TPTM, exploiting temporal dependencies and user interactions, shows superior performance compared to Sembler.



(a) Held-out log-likelihood of three methods.

(b) Held-out log-likelihood with different γ_c.

Figure 5: Held-out log-likelihood analyses

Log-likelihood with respect to the number of topics K is shown in Table 5. We trained TPTM by 25, 50, 75, 100 topics respectively with the best at 25. TPTM outperforms both Sembler and LDA consistently and significantly, which shows the power of modeling temporal and personalized factors in TPTM. We also observe that in both datasets, log-likelihood decreases when the number of topics increases. This may be due to the data we used. For example, music video users mainly appreciated hot songs and therefore topics were limited. For fun videos, although the semantics of different videos vary, the main topics may be limited to only a few such as daily life and pets. In practice, K can be optimized using cross-validation or non-parametric methods.

We also studied γ_c which decides how much a comment is affected by its preceding comments. We tested γ_c from 0.1 to 0.9, and the result is shown in Figure 5(b). When γ_c was set to 0.3, TPTM performed the best. Referring to Equation 3, when $\gamma_c = 0.3$, the temporal effect of the current comment semantics decays by 25% after one comment, and 78% after five comments.

4.4 User Study

We have also conducted a set of user studies (Figure 6) to evaluate the quality of tags generated by TPTM compared to Sembler

[14]For example, 'high energy' is a Chinese internet slang often used to forecast the coming eye-catching video event.

Table 4: Top words for four selected topics for each model ($K = 100$).

Method	Topic 1	Topic 2	Topic 3	Topic 4
TPTM	'BGM[15]', 'Tacata', 'download'	'high energy', 'high energy ahead', 'low energy'	'divine tune', 'good', 'jazz'	'style', 'oba', 'gangnam'
Sembler	'music', 'miku', 'dating'	'high energy', 'high energy ahead', 'low energy'	'divine tune', 'hungry', 'awesome'	'style', 'oba', 'pickled vegetables'
LDA	'BGM', 'train', 'translation'	'ice', 'high energy', 'low energy'	'divine tune', 'little hero', 'monster'	'oba', 'I cry', 'gangnam'

Table 5: Held-out log-likelihood performance ($\times 10e7$) with different number of topics K.

Method \ K	25	50	75	100
Data: Music Videos				
TPTM	**-2.269 ± 9e-5**	**-2.2816 ± 5e-5**	**-2.2884 ± 3e-5**	**-2.2939 ± 5e-5**
Sembler	-2.2714 ± 8e-5	-2.2826 ± 8e-5	-2.2892 ± 2e-5	-2.2948 ± 6e-5
LDA	-2.2746 ± 9e-5	-2.2859 ± 6e-5	-2.2925 ± 3e-5	-2.2974 ± 5e-5
Data: Fun Videos				
TPTM	**-3.8995 ± 3e-5**	**-3.9227 ± 4e-5**	**-3.9359 ± 4e-5**	**-3.9448 ± 4e-5**
Sembler	-3.9021 ± 4e-5	-3.9242 ± 4e-5	-3.9370 ± 4e-5	-3.9460 ± 4e-5
LDA	-3.9084 ± 3e-5	-3.9302 ± 3e-5	-3.9424 ± 3e-5	-3.9511 ± 4e-5

and LDA. Three labelers evaluated the randomly selected 673 shots from the music videos.

Figure 6: User study interface.

As illustrated in Figure 6, shots were played one by one and the corresponding tags generated by the three methods were displayed on the right-hand-side. Specifically, the order of the three sets of tags were random for each shot. The number of tags in each set was at most ten and the tag order represents its ranking and relevance. Labelers were asked to choose the best set of tags by clicking on one of the three radio buttons in the right-hand-side. One of the methods then received a single vote for each shot. The results listed in Table 6 show that TPTM received 28% more votes than the other methods. Fleiss' Kappa [10], a measure of inter-rater reliability, was then evaluated. The Fleiss' Kappa of the votings was 0.20 with a p-value at 10^{-4}, which can be interpreted as fair but statistically significant agreement among the labelers.

Table 6: Results of user study.

Method	A	B	C	Overall
TPTM	267	275	240	782
LDA	207	192	209	608
Sembler	179	186	204	569

4.5 Case Study

We randomly picked a video[16] and examined its time-sync tags. As listed in Table 7, we extracted six shots of the tags from the video and listed the corresponding snapshots in the first row. Words below the snapshots are generated by the three methods. Note that words are in descending order of relevance, and words in bold are manually labeled as meaningful tags. In general, all three methods generated some meaningful words from the *short* and *noisy* comments. TPTM provided a more reasonable rankings for these tags. Moreover, TPTM captured more valuable words given the *short* and *noisy* content, such as 'good looking' in shot 2, 'princess' in shot 4 and 'American' in shot 6. More specifically, the mean average precision at ten tags (MAP@10), a standard evaluation metric of ranking, was calculated. TPTM achieved 0.187, which is 30% higher than Sembler and LDA (0.146 and 0.144 respectively).

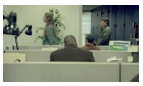
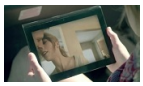
[16]http://www.acfun.tv/v/ac268521

5. RELATED WORK

Traditional video tagging techniques generate tags for an entire video clip [17, 20]. Some content-based methods for time-sync tagging have been proposed (e.g., Feng *et al.* designed a model to generate time-sync tags by predicting tags for extracted keyframes [9]). However, content-based methods rely on a large amount of human-labeled data, which is difficult to acquire in real-world applications because the types of videos vary widely and human labor is expensive. Some researchers have turned to the cheap or free data acquired from the internet. For example, Xu *et al.* and Chiu *et al.* used web-casting text to detect events in broadcast videos [24, 6]. Chakrabarti *et al.* [5] summarized live sports videos events using user-generated information in social networks, the output of which, however, are multiple key tweets instead of a few tags. Also, the above methods mainly focused on big events such as sports games, which cannot be extended to daily online videos. Davis *et al.* designed a system to study time-sync tagging by social network users [8]. However, their data was acquired from an experimental system, which is not scalable. To the best of our knowledge, no previous work has been done on text based time-sync video tagging, and none has solved the *short* and *noisy* comment problem of crowdsourced content.

Crowdsourcing is a process that involves outsourcing tasks to a distributed group of people, which is normally much cheaper than hiring experts. Machine learning and data mining researchers have been using crowdsourcing services (e.g., Amazon Mechanical Turk) to solve the lack of labeled-data problem for applications such as sentiment classification [18] and Name Entity Recognition [23]. However, crowdsourced labels need to be cleaned before utilization because labels from the crowd are usually contaminated by

Table 7: Example of time-sync video tagging. Description of the six segments: 1) At the beginning of the video, Taylor Swift showed up as an office lady. 2) Taylor Swift is working in an office. 3) A man is playing card games in the computer. 4) Taylor Swift is taking the elevator. 5) Taylor Swift is recalling the moments with her boyfriends using a Sony tablet. 6) Taylor Swift picks up her soldier boyfriend at the airport.

Shots	31.3"-45.9" 	105.8"-116.0" 	149.8"-153.4" 	161.0"-172.0" 	178.4"-190.8" 	235.7"-241.6" 	MAP@10
TPTM	'subtitles' '+1' **'like'** ',,' 'mine' **'love her'** 'you' 'really' **'big love'** **'my love'**	'.....' **'divine tune'** **'like'** 'ah..' 'holy cow' 'the' **'MV'** **'sister'** 'pressure' **'good looking'**	'have to' **'game'** 'to' 'have to be' 'just now' **'memory'** **'computer'** **'on the screen'** 'what' 'and'	'mine' 'you' **'sister'** 'have to' 'ah,' 'it's me' 'again' 'a little' **'singer'** **'princess'**	'you' 'the' **'I love'** 'in fact' 'yours' 'my' **'love you'** 'truth' **'ads'** **'expression'**	'feeling' **'girl'** 'have to' **'bastard'** 'all are' **'ads'** **'beautiful'** 'is?' **'man'** **'American'**	0.187
Sembler	'subtitles' '+1' **'like'** ',,' 'mine' 'no' 'really' 'think' 'OK' 'you'	'.....' 'holy cow' **'divine tune'** **'like'** 'the' **'MV'** **'sister'** **'I like'** 'pressure' 'ah..'	'have to' **'game'** 'to' 'just now' 'have to be' **'computer'** **'memory'** **'on the screen'** 'what' 'and'	'mine' 'again' 'you' **'sister'** 'ah,' 'it's me' 'have to' 'a little' 'people' **'singer'**	'you' **'I love'** 'the' 'yours' 'in fact' 'truth' **'love you'** 'my' **'expression'** **'ads'**	'feeling' **'girl'** 'all are' 'have to' 'is?' **'man'** 'is,' **'ads'** **'bastard'** **'beautiful'**	0.146
LDA	'+1' 'subtitles' **'like'** ',,' 'mine' 'really' 'you' 'I will' **'I love'** **'love her'**	'.....' **'divine tune'** **'like'** 'holy cow' **'sister'** 'the' 'pressure' 'ah..' **'MV'** 'great'	'have to' **'game'** 'to' 'just now' 'have to be' **'memory'** **'computer'** **'on the screen'** 'what' 'and'	'mine' 'have to' 'you' **'sister'** 'again' 'ah,' 'a little' 'it's me' 'people' **'singer'**	'you' 'the' **'I love'** 'yours' 'truth' 'in fact' 'my' **'love you'** **'expression'** **'ads'**	'feeling' **'girl'** 'all are' 'have to' **'man'** 'is?' **'ads'** **'bastard'** **'beautiful'** 'is'	0.144

errors and bias. Several approaches have been proposed recently for label by emphasizing labels provided by high quality labelers cleaning[3, 22, 23]. For example, Welinder *et al.* [22] designed a probabilistic graphical model to evaluate skill and knowledge for each image annotator, as well as quality for each image. Sembler [23] was proposed to improve the quality of the collected labels in a sequential labeling problem by modeling users' abilities and sequential dependencies of instances. In summary, most traditional methods infer the true label for a given instance by modeling both user ability and the question difficulty, assuming that users answer questions independently. However, user labeling does not follow the *i.i.d.* assumption in TSC videos due to users' interaction in the shared watching experience. Therefore, traditional methods may fail to accurately remove users' bias. Although Das *et al.* addressed the user interaction problem in a crowdsourcing setting [7], only single-real-value label space and explicit social network were considered. These are not applicable to a video tagging/topic extraction problem. Ritter *et al.* modeled twitter dialogues, another type of interacting short message, by assuming fixed topics, and homogeneous users [16]. Their model, however, is essentially a similar, simpler version of Sembler and TPTM for a different purpose (discovering dialogue acts).

Time-sync commented videos have been exploited by Yoshii *et al.*. The authors developed an automatic music commentator using TSC videos [25]. Their method, however, failed to work in our problem setting because once their model is trained, the most probable words for each hidden state are fixed. This limits the vocabulary of generated comments or tags, which can be observed from their online demo[17]. Moreover, they did not address the *short* and *noisy* comment problems.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have exploited crowdsourced texual data from time-sync commented video websites for automatic time-sync video tagging, which is a new application. Based on the challenge that time-sync comments in each video shot are *short* and *noisy*, we proposed a novel temporal and personalized topic model which enriches knowledge of *short* comments across videos and shots by collectively exploiting multiple users' preferences. It also peels off user interactions in time-sync comments to address the *noisy* comment problem. Held-out log-likelihood analyses and user studies show that our proposed model outperforms state-of-the-art baselines. Case studies have also been conducted to demonstrate our proposed model's capability in mining *short* and *noisy* comments.

Discussion. According to Section 3.3, to train the model (i.e., to infer variables such as user preference x and the topic distribution of a shot $\mathcal{T}$), we need to scan through all the data. This can be a problem in tagging online videos where video collections are large scale and growing, which makes the training process very slow. Our model can be easily modified into an online and paral-

[17]http://staff.aist.go.jp/k.yoshii/commentator/index.html

lel scheme to handle large scale data. For updating the lower level variables (e.g., θ and φ), parallel and online LDA have been proposed for efficient topic inference [12]. And for updating higher level variables (e.g., λ and x), the user preference x is the only factor learned from the old data that relates to topics of the new videos. Therefore, we only need to load user preference x learned in the old data to absorb new data, which makes online updating trivial. Again, since variables such as λ, m_{pre_s}, and m_{pre_c} only depend on the corresponding video, the updating process can be easily partitioned with respect to each video, that is to update multiple videos in parallel. The online and parallel scheme is essential for handling large scale data (though the details are not elaborated here due to page limits). In addition, although our proposed model is specifically designed for time-sync tagging of online videos, it can also serve as a basic technique for other applications such as time-sync tagging of news video and movies with the help of subtitles or speech-to-text transcription. Moreover, our proposed pure text-based model can also be applied to many tasks other than video tagging such as event summarization based on social networks.

Future Work. In the future, we can extend our work as follows:

1) By building internet slang collections to improve tokenization. According to Table 7 and our observations, the quality of the generated tags heavily depends on pre-processing such as tokenization. However, to the best of our knowledge, no large internet slang collections are available. Although we have not analyzed TSC from a linguistic point of view due to the limits of time and our knowledge, we consider it the most important area for future work.

2) By designing a unified model incorporating video segmentation. Video segmentation is essential to the user-experience of time-sync tagging. However, to simplify the problem, video segmentation has been conducted by a simple segmentation scheme as a part of data pre-processing in this paper, which is not satisfactory in terms of quality. In the future, we can incorporate video segmentation as a latent factor, and infer better segmentation simultaneously considering the semantics of the comments .

3) By improving tagging quality by knowledge transfer among multiple data sources. In this paper, connection between different TSC video sources was not considered. In fact, videos in different sections can be very different in terms of both video semantics and users' comments. For example, videos in the music section usually consists of multiple short clips, while videos in the movie section are likely much longer in terms of per-video duration. The underlying topics are probably different as well. It would be interesting to investigate how users comment in different sections.

4) By extending our results to extrinsic tasks such as video scene classification and object recognition. The automatic time-sync tagging approach proposed in this paper is actually a crowdsourced label collection and integration process. Therefore, a natural extension is to use the generated tags and the corresponding video shots as labeled pairs for extrinsic tasks. On the other hand, the quality of the generated tags can also be evaluated by existing ground-truth measures of extrinsic tasks.

7. ACKNOWLEDGMENTS

This work was undertaken thanks to the support of China National 973 project 2014CB340304 and Hong Kong RGC Projects 621013, 620812, and 621211.

8. REFERENCES

[1] Cisco visual networking index: Global mobile data traffic forecast update, 2012-2017. white paper. Online: http://goo.gl/wt9Dy (9 July 2013).

[2] Youtube statistics. Online: http://goo.gl/jLcaFP (9 July 2013).

[3] Y. Bachrach, T. Graepel, T. Minka, and J. Guiver. How to grade a test without knowing the answers - a bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *ICML*, 2012.

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

[5] D. Chakrabarti and K. Punera. Event summarization using tweets. In *ICWSM*, pages 66–73, 2011.

[6] C. Chiu, P. Lin, S. Li, T. Tsai, and Y. Tsai. Tagging webcast text in baseball videos by video segmentation and text alignment. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(7):999–1013, 2012.

[7] A. Das, S. Gollapudi, R. Panigrahy, and M. Salek. Debiasing social wisdom. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 500–508. ACM, 2013.

[8] S. Davis, I. Burnett, and C. Ritz. Using social networking and collections to enable video semantics acquisition. *IEEE MultimediaS*, 16(4):52–60, 2009.

[9] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, pages 1002–1009, 2004.

[10] J. L. Fleiss, B. Levin, and M. C. Paik. The measurement of interrater agreement. *Statistical Methods for Rates and Proportions*, 2:212–236, 1981.

[11] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[12] D. K. JinYeong Bak and A. Oh. Distributed online learning for latent dirichlet allocation. In *NIPS Workshop on Big Learning, 2012*, pages 1–8, 2012.

[13] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *RecSys*, pages 61–68. ACM, 2009.

[14] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*, volume 999. MIT Press, 1999.

[15] D. M. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, pages 411–418, 2008.

[16] A. Ritter, C. Cherry, and B. Dolan. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, 2010.

[17] S. Siersdorfer, J. San Pedro, and M. Sanderson. Automatic video tagging using content redundancy. In *SIGIR*, pages 395–402, 2009.

[18] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*, pages 254–263, 2008.

[19] S. Tan, H. Tan, and C. Ngo. Topical summarization of web videos by visual-text time-dependent alignment. In *MM*, pages 1095–1098. ACM, 2010.

[20] A. Ulges, C. Schulze, M. Koch, and T. Breuel. Learning automatic concept detectors from online video. *Computer Vision and Image Understanding*, 114(4):429–438, 2010.

[21] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th ICML*, pages 1105–1112. ACM, 2009.

[22] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, pages 2424–2432, 2010.

[23] X. Wu, W. Fan, and Y. Yu. Sembler: Ensembling crowd sequential labeling for improved quality. In *AAAI*, pages 1713–1719, 2012.

[24] C. Xu, J. Wang, K. Wan, Y. Li, and L. Duan. Live sports event detection based on broadcast video and web-casting text. In *Proceedings of the 14th ACM MM*, pages 221–230. ACM, 2006.

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