Personalized Key Frame Recommendation

Xu Chen*
Tsinghua University
xu-ch14@mails.tsinghua.edu.cn

Yongfeng Zhang*
University of Massachusetts Amherst
zhangyf07@gmail.com

Qingyao Ai
University of Massachusetts Amherst
aiqy@cs.umass.edu

Hongteng Xu
Georgia Institute of Technology
hxu42@gatech.edu

Junchi Yan
East China Normal University
IBM Research –– China
jcyan@sei.ecnu.edu.cn

Zheng Qin†
Tsinghua University
qinzh@mails.tsinghua.edu.cn

ABSTRACT

Key frames are playing a very important role for many video applications, such as on-line movie preview and video information retrieval. Although a number of key frame selection methods have been proposed in the past, existing technologies mainly focus on how to precisely summarize the video content, but seldom take the user preferences into consideration. However, in real scenarios, people may cast diverse interests on the contents even for the same video, and thus they may be attracted by quite different key frames, which makes the selection of key frames an inherently personalized process. In this paper, we propose and investigate the problem of personalized key frame recommendation to bridge the above gap. To do so, we make use of video images and user time-synchronized comments to design a novel key frame recommender that can simultaneously model visual and textual features in a unified framework. By user personalization based on her/his previously reviewed frames and posted comments, we are able to encode different user interests in a unified multi-modal space, and can thus select key frames in a personalized manner, which, to the best of our knowledge, is the first time in the research field of video content analysis. Experimental results show that our method performs better than its competitors on various measures.

KEYWORDS

Key Frame; Personalization; Recommender Systems; Collaborative Filtering; Video Content Analysis

1 INTRODUCTION

All along, key frames are of fundamental importance for the video-based applications, for example, users in the on-line movie websites can readily preview a video under the help the exhibited key frames even without watching the whole content, video search engines can efficiently return the results by matching the queries with the key frames of the candidate items. Although key frame extraction methods [10, 22, 58] have been widely investigated in the past, the proposed technologies mostly aim to summarize the video content, but seldom consider user preferences on the extracted key frames. However, in many applications, different people may be interested in various video contents, and thus may be attracted by quite different key frames. Without the guidance of tailored key frames, users may easily miss their potentially favorite videos. Therefore, in this paper, we would like to ask "whether it is possible to design an effective model to select and recommend personalized key frames according to users' different tastes."

In real scenarios, the main challenge to answer the above question is the lack of users' personalized interaction information that reveals their "frame-level" viewing preferences. Fortunately, the emerging of video sharing websites such as Niconico[1], BiliBili[2], and AcFun[3] may shed light on this problem, where users are allowed to express opinions directly to the frames of interest by time-synchronized comments (or TSCs, first introduced in [49], see Figure 1) in a real-time manner. Intuitively, the user behaviors of commenting on a frame can be regarded as implicit feedback reflecting the *frame-level preference*, while the image features of the reviewed frame and the text features in the posted time-synchronized comment can further help to model the *user specific (or finer-grained) preference* from different perspectives. For example in Figure 1, user A expresses her preference on a frame with time-synchronized comment "*... I like his overcoat, it looks cool and also must be very comfortable with good quality*". On one hand, from the content of the time-synchronized comment we can indicate that the user express a positive sentiment on this frame, and the particular aspect that attracts her attention is the quality of the clothes. On the other hand, from the frame image, we can further infer the visual features of her interests, such as clothing texture, which are usually difficult to describe with text. By leveraging all the historical implicit feedback as well as the features (image and text) of users' interest, we can collaboratively match a target user with her potentially favorite frames.

Based on the above motivation and intuition, we describe and analyze a novel **K**ey **F**rame **R**ecommender by modeling user time-synchronized **C**omments and the key frame **I**mages simultaneously (called **KFRCI**). The main building block of our proposed method is to integrate the power of model-based collaborative filtering and

* Both authors contributed equally to this work.
† Corresponding author.

SIGIR '17, August 7–11, 2017, Shinjuku, Tokyo, Japan
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.
978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080776

[1] http://www.nicovideo.jp
[2] http://www.bilibili.com
[3] http://www.acfun.cn

long-short term memory network. The carefully designed collaborative filtering component aims to capture user preferences based on image features, while the modified long-short term memory network component aims to model user time-synchronized comments to excavate her personalized opinions toward different frames. Furthermore, by integrating these two components, we build a unified framework that can encode user preference in a multi-modal space so as to facilitate comprehensive user profiling and accurate key frame recommendation.

Compared with previous works, the main contributions of our paper are as follows:

- We propose and investigate the problem of personalized key frame recommendation, and to the best of our knowledge, this is the first work to select video key frames based on users' personalized preferences.
- For better addressing the above novel problem, we present a novel neural architecture that combines collaborative filtering and long-short term memory network together to jointly model user time-synchronized comments and video key frame images.
- We conduct extensive experiments to demonstrate the effectiveness of our proposed models, and also we present detailed analysis on the parameters as well as the effects of different information sources (image and text) in our framework.

In the rest of the paper, we first introduce the related work in section 2, and then formally define our problem in section 3. Our framework is illustrated in section 4. In section 5, we verify the effectiveness of our method with experimental results. Conclusions and outlooks of this work are presented in section 6.

2 RELATED WORK

2.1 Review-based Recommendation

Recommender system is a well studied field with many effective models proposed [5, 7, 17, 19, 35, 39]. Recently, for better capturing user preferences, user reviews has attracted more and more research interest [3, 42, 43], and many review-based models have been proposed to improve the recommendation performance [2, 4, 9, 11, 29, 31, 37, 41, 45, 46, 51, 56] or enhance the interpretability [16, 38, 50, 50, 57].

According to the review processing methods, these models can be generally classified into two categories. On one hand, some methods leverage the review text on document- or review-level, which take every piece of user review as a whole for global analysis. Specifically, [31, 41] link the latent factors in rating data with the topics in the textual review to generate more accurate recommendations, and [9, 51] propose to leverage probabilistic graphical method to include more flexible prior knowledge for review modeling. To better capture the local semantic information in user reviews, [56] combine traditional matrix factorization technology with word2vec [34] for more precise review modeling and recommendation.,

On the other hand, some approaches try to leverage textual reviews on a feature- or aspect-level, which first extract product features and user sentiments from user reviews, and then represent the unstructured free-text reviews as structured feature-opinion pairs to facilitate finer-grained user preference modeling. Particularly, [57] use multi-matrix factorization to generate explainable recommendations based on the extracted product features. [4] further captures user favorite product features in a learning to rank manner.

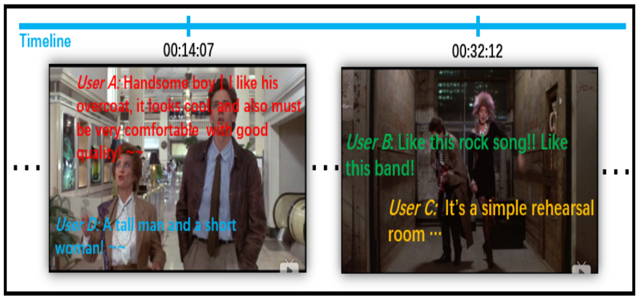
Timeline
00:14:07
00:32:12
User A: Handsome boy I I like his overcoat, it looks cool, and also must be very comfortable with good quality! ~~
User D: A tall man and a short woman! ~~
User B: Like this rock song!! Like this band!
User C: It's a simple rehearsal room ···
...
...
...

Figure 1: A simple example of TSC. Different users may express real-time opinions directly upon their interested frames. The comments are manually translated into English by the authors.

Compared with the aforementioned methods, we leverage long-short term memory network in our model to capture the word sequential properties mirrored in the time-synchronized comments, which has mostly been ignore by the previous works.

2.2 Image-based Recommendation

Recently, there is a trend to incorporate visual features into the research of personalized recommendation [8, 15, 32]. Specifically, [15] infuses the image features into the traditional ranking-based method to improve the performance of Top-N recommendation. [8] combines the product images and item descriptions together to make dynamic predictions. [32] leverages visual features to discover different relationships between items.

Generally, these methods aim to take advantage of image features to boost the performance of traditional item recommendation, such as product recommendation in E-commerce. Instead, we aim at a very different task of key frame (which itself is an image) recommendation, where image is not a side information but the target to process. Besides, previous works did not capture user preference from the time-synchronized comments, which is another main difference when compared with our model.

2.3 Time-synchronized Comments

Time-Synchronized Comments (TSC) is first introduced in [49] for automatic video shot tagging, where the authors proposed a novel method to extract time-synchronized video tags by automatically exploiting crowd-sourcing comments. [52] further leverages TSC to extract highlight shots for a video with a frequency-based method. However, the extracted highlight shots are static and could not provide tailed key frames for different users, which is the inherent difference from the personalized key frame recommendation task targeted in our work.

2.4 Key Frame Extraction

A similar research direction to our work is key frame extraction (or video summarization), it has attracted much research interest and many models have been proposed in the past. Early works [13, 28] usually extract visual features of frames at first, and then cluster frames accordingly to generate key frames. To improve the performance, other types of information beyond visual features are considered in recent work, including the viewer attention [30, 53, 54], audio signal [23], subtitles [27], etc. Moreover, semantic information has also been exploited to summarize videos, including special events [47, 48], key people and objects [24, 26], and story-lines [25]. Compared with these models, our work essentially make a further step forward by distinguishing user preferences on the extracted key frames.

3 PRELIMINARIES AND PROBLEM DEFINITION

3.1 Dataset Inspection

The data used in our work is crawled from a well-known video sharing website Bilibili[4]. We obtained the time-synchronized comments from the movie category till December 10th, 2015. To better understand the insights of this dataset, we conducted preliminary statistic analyses, which are listed as Table 1 and Figure 2.

Table 1: Overall statistics of the time-synchronized comment (TSC) dataset.

Total number of users (#users)	1,133,750
Total number of movies (#movies)	7,166
Total number of TSCs (#TSCs)	11,842,166
Ave. TSCs per movie	1,652.5
Ave. users per movie	465.9
Ave. TSCs per user	10.4
Max/Min number of TSCs for a movie	8,028/101
Max/Min number of users for a movie	3,370/1
Max/Min number of TSCs for a user	68,236/1

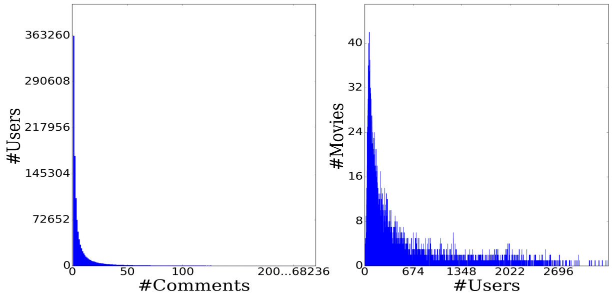

Figure 2: On the left is the relation between the number of comments and users, while on the right is the relation between the number of users and movies.

As can be seen in Figure 2, the quantity of active users is relatively small, and most users only sent a small number of TSCs, which conforms to the "long tail" theory that is frequently observed in user behavior analysis. Similar results can also be found between the number of users and movies.

3.2 Problem Formalization

Suppose there are N users $\boldsymbol{u} = \{u_1, u_2, ..., u_N\}$ and M videos $\boldsymbol{v} = \{v_1, v_2, ..., v_M\}$. We first segment each movie $v_i \in \boldsymbol{v}$ into L shots $\boldsymbol{l}_{v_i} = \{l_{v_i}^1, l_{v_i}^2, ..., l_{v_i}^L\}$, and then generate key frames correspondingly leveraging existing methods such as [33, 36]. The key frame in shot $l_{v_i}^j$ is defined as $k_{v_i}^j$, and $\boldsymbol{K}$ represents the whole set of key frames among all the videos.

Suppose the visual features of key frame k is defined as $\boldsymbol{vsl}_k$, and let user u's time-synchronized comment on key frame k be tsc_{uk} with sentiment polarity pol_{uk}, which is determined by the Stanford sentiment analysis toolkit[5]. The word list in tsc_{uk} is defined as $\boldsymbol{w}_{tsc_{uk}} = \{w_{tsc_{uk}}^0, w_{tsc_{uk}}^1, ..., w_{tsc_{uk}}^{s_{uk}-1}\}$, where s_{uk} is the length of the comment.

Given all the visual features $\boldsymbol{VSL} = \{\boldsymbol{vsl}_k | k \in \boldsymbol{K}\}$ as well as users' historical time-synchronized comments $\boldsymbol{W} = \{\boldsymbol{w}_{tsc_{uk}} | u \in \boldsymbol{u}, k \in \boldsymbol{K}\}$ and their corresponding sentiments $\boldsymbol{POL} = \{pol_{uk} | u \in \boldsymbol{u}, k \in \boldsymbol{K}\}$, for a target user u and one of her unseen movie $v_i \in V$ with preselected key frames $\{k_{v_i}^1, k_{v_i}^2, ...k_{v_i}^L\}$, our task is to find a function $g(\cdot)$ to re-rank these key frames according to $u's$ interest, that is, $g(\{k_{v_i}^1, k_{v_i}^2, ..., k_{v_i}^L\} | u, \boldsymbol{W}, \boldsymbol{POL}, \boldsymbol{VSL}) = \{k_{v_i}^{o1}, k_{v_i}^{o2}, ...k_{v_i}^{oL}\}$, where $\{o1, o2, ..., oL\}$ is an ordering of $\{1, 2, ..., L\}$. The top n key frames among the final results are at last recommended (shown) to user u. To make more clear presentation, we list the notations used throughout the paper in Table 2.

4 KEY FRAME RECOMMENDER

We first propose an improved collaborative filtering method to capture users' frame-level preference by making use of image visual features. Then to model the textual features mirrored in time-synchronized comments, we modify the long short term memory network by infusing per-user personalization information. Lastly, we design a unified framework to jointly model frame images as well as time-synchronized comments.

For clarity and integrality, we first re-describe the widely used matrix factorization (MF) model as a neural network. Formally, let $\boldsymbol{p_u}$ and $\boldsymbol{q_k}$ represent the latent factors of user u and item k, then the likeness (or score) of u to k can be predicted as $\hat{y}_{uk} = \boldsymbol{p_u}^T \boldsymbol{q_k}$. In the context of neural network (see Figure 3), the user/item IDs with one-hot format can be seen as inputs fed into the architecture, then the embedding layer projects these sparse representations into denser vectors, which can be regarded as the latent factors in matrix factorization models. At last, the final result $\hat{y}_{uk}$ is computed as the vector inner product between $\boldsymbol{p_u}$ and $\boldsymbol{q_k}$.

4.1 Image-based Model

To capture user preference from frame images, we fuse the visual features into the above framework. Specifically, our principled design is shown in Figure 4. Suppose the dimension of the user/frame

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

[5] https://nlp.stanford.edu/sentiment/

Table 2: Notations and descriptions.

Notations	Descriptions
$\boldsymbol{u}$	The set of N users $\{u_1, u_2, \ldots, u_N\}$.
$\boldsymbol{v}$	The set of M movies $\{v_1, v_2, \ldots, v_M\}$.
$\boldsymbol{l}_{v_i}$	The set of L shots $\{l_{v_i}^1, l_{v_i}^2, \ldots l_{v_i}^L\}$ for movie v_i.
$k, k_{v_i}^j, \boldsymbol{K}$	An arbitrary key frame, the key frame of shot $l_{v_i}^j$, and the set of all key frames.
$\boldsymbol{vsl}_k, VSL$	The preprocessed visual feature of frame k, and the set of all visual features.
tsc_{uk}, pol_{uk}	u's time-synchronized comment on key frame k, and the polarity of tsc_{uk}
$\boldsymbol{w}_{tsc_{uk}}, \boldsymbol{W}$	The word list $\{w_{tsc_{uk}}^0, w_{tsc_{uk}}^1, \ldots w_{tsc_{uk}}^{s_{uk}-1}\}$ in tsc_{uk}, and the set of all time-synchronized comments.
$\boldsymbol{p_u}, \boldsymbol{q_k}$	Latent factors of user u and frame k.
O^+, O^-	The set of positive, and sampled negative feedbacks.
K^{neg}	Number of negative instances.
N^{word}	Size of the word vocabulary.
$\boldsymbol{h_t}$	The hidden state in LSTM at iteration t
D	The number of non-linear layers.
$\boldsymbol{e_{uk}^{pre_0}}, \boldsymbol{e_{uk}^{pre_1}}, \ldots, \boldsymbol{e_{uk}^{pre_D}}$	Preference embeddings of u on k.
$\boldsymbol{W^{image}}, \boldsymbol{W^{i}}, \boldsymbol{w^{output}}$	Weighting matrix that maps $\boldsymbol{vsl}_k$ into a K dimensional vector, the coefficient matrix used to weight $\boldsymbol{e_{uk}^{pre_{i-1}}}$, and a vector that maps $\boldsymbol{e_{uk}^{pre_D}}$ into a scalar.
$MERGE(), LOGISTIC(), LSTM()$	The merge function, logistic function, and LSTM network.
$\hat{y}_{uk}^{image}, \hat{y}_{uk}^{TSC}, \hat{y}_{uk}^{integrated}$	User u's predicted likeness score to frame k when using image, TSC, and integrated information.

latent factors (embedding) is d, then each visual feature is first mapped into a d dimensional vector, which is then merged with the frame latent factor to generate a new embedding (blue circle in the figure). Lastly, the user latent factors together with the newly generated embedding are fed into the inner product layer to compute the final prediction.

Image enhanced key frame representation. Again, let $\boldsymbol{p_u}$ and $\boldsymbol{q_k}$ be the latent factors of user u and key frame k, respectively. We first use Caffe deep learning framework [21] to generate visual features from the original frame images, where we adopted the Caffe reference model with 5 convolutional layers followed by 3 fully-connected layers that has been pre-trained on 1.2 million ImageNet (ILSVRC2010) images. For frame k, we use the output of FC7, namely, the second fully-connected layer, as the final visual feature $\boldsymbol{vsl}_k$, which is a feature vector of length 4096.

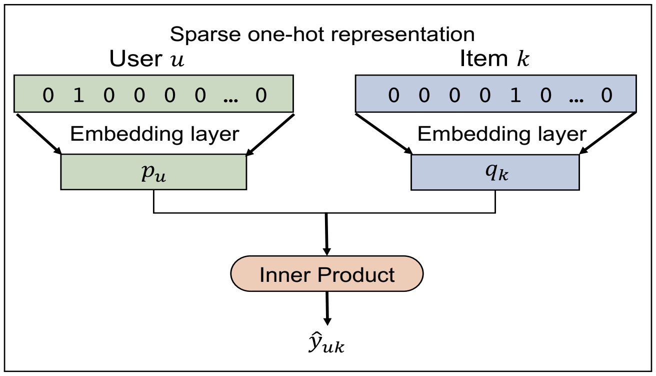

Figure 3: Matrix Factorization as Neural Network.

Let $\boldsymbol{W^{image}} \in R^{d \times 4096}$ be the weighting matrix that maps $\boldsymbol{vsl}_k$ into a d dimensional vector, then the new key frame representation can be derived as:

$$\boldsymbol{q_k^*} = MERGE(\boldsymbol{q_k}, \boldsymbol{W^{image}} \cdot \boldsymbol{vsl}_k) \tag{1}$$

where $MERGE : R^d \times R^d \to R^d$ is a function that merges two d dimension vectors into one. The particular choice of $MERGE$ in our model is a simple element-wise multiplication, i.e.,

$$MERGE\Big((a_1, a_2, ..., a_K), (b_1, b_2, ..., b_K)\Big) = (a_1 b_1, a_2 b_2, ..., a_K b_K) \tag{2}$$

however, it is not necessarily restricted to this function and many choices can be used in practice according to the specific application scenario.

Sentiment-based user preference modeling. Intuitively, users would comment on their favorite frames with positive sentiment. Therefore, we first determine the polarity of each time-synchronized comment by the Stanford Sentiment analysis toolkit[6], and then for simplicity, we set the polarity of tsc_{uk} – i.e., pol_{uk} – as 1 if the result is *very positive*, *positive*, *or neutral*, and 0 otherwise.

In our framework, we take the prediction of a user's likeness to a frame as a binary classification problem, where 1 means a user likes a frame, and 0 otherwise; the likeness of user u to frame k – i.e., $\hat{y}_{uk}^{image} \in [0, 1]$ – can be predicted as:

$$\hat{y}_{uk}^{image} = LOGISTIC(\boldsymbol{p_u} \cdot \boldsymbol{q_k^*}). \tag{3}$$

where $LOGISTIC(x) = \frac{1}{1+e^{-x}}$ is the logistic function, and "·" denotes inner product.

Then we use the binary cross-entropy as our loss function to model user preference, whose superiority has been explored and demonstrated in [18], and the final objective function to be maximized is:

[6] https://nlp.stanford.edu/sentiment/

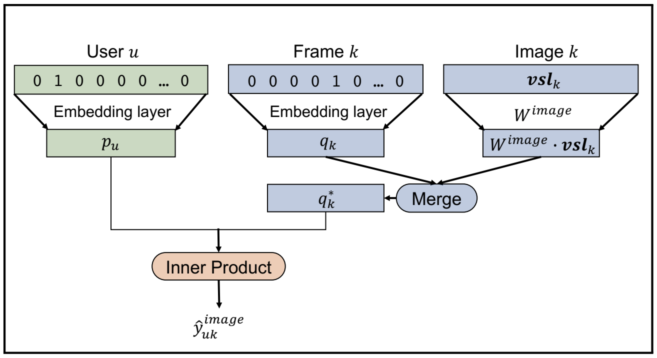

Figure 4: The framework of image-based model. The preprocessed image feature is merged with frame latent factors to derive a new embedding, which is then multiplied by the user latent factors to generate the prediction.

$$
\begin{aligned}
L_1 &= log \prod_{(u,k)} (\hat{y}_{uk}^{image})^{y_{uk}} (1 - \hat{y}_{uk}^{image})^{1-y_{uk}} \\
&= log \prod_{(u,k)\in O^+} \hat{y}_{uk}^{image} \prod_{(u,k)\in O^-} (1 - \hat{y}_{uk}^{image}) \\
&= \sum_{(u,k)\in O^+} log\, \hat{y}_{uk}^{image} + \sum_{(u,k)\in O^-} log\, (1 - \hat{y}_{uk}^{image})
\end{aligned}
\tag{4}
$$

where y_{uk} is the ground truth that would be 1 if u has commented on k with $pol_{uk} = 1$, and 0 otherwise. O^+ is the set of positive feedbacks, i.e., $O^+ = \{(u,k)|u\ has\ commented\ on\ k, pol_{uk} = 1\}$, while O^- is the set of sampled negative feedback, namely, $O^- \subseteq O^{neg} \cup O^*$, where $O^{neg} = \{(u,k)|u\ has\ commented\ on\ k, pol_{uk} = 0\}$, and $O^* = \{(u,k)|u\ has\ not\ commented\ on\ k\}$. In the training phase, for each positive feedback (u,k), we uniformly sample K^{neg} negative instances, and the parameters can be learned via stochastic gradient descent (SGD).

4.2 Text-based Model

Existing review-based recommendation methods mostly consider the words in a comment as independent elements, and they usually ignore the word sequential information – which is yet very important for understanding the semantic of a comment. In Figure 1 for example, user D wrote the review "*A tall man and a short woman*", where if we leave out the consideration of word sequential information, it would be computationally identical to "*A tall woman and a short man*", which obviously expresses a completely opposite meaning.

To capture the word sequential information, we make use of the long short term memory (LSTM) [20] network, which has been successfully applied to a number of sequence modeling tasks such as machine translation [1], image caption [44], and video classification [55].

Preference-aware LSTM. Intuitively, the content of a time-synchronized comment on a frame is influenced by both the user preference and the frame itself. When it comes to our model, as a result, the word generation process in LSTM should be influenced by both the user and the frame latent factors. So in our framework as shown in Figure 5, we first merge the user and frame latent factors into a preference embedding using element-wise vector multiplication, and then feed it as an extra input to LSTM at each step.

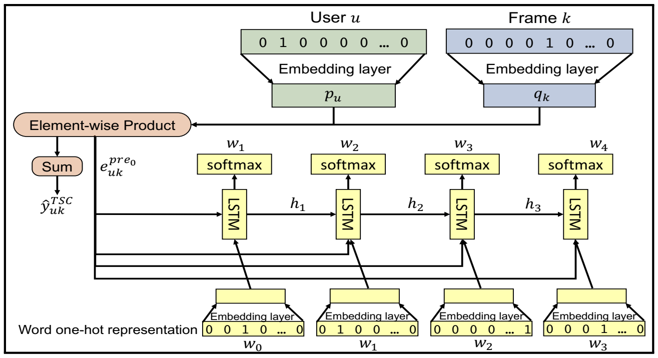

Figure 5: The framework of text-based model. The preference embedding $\boldsymbol{e_{uk}^{pre_0}}$ is fed as an extra input to LSTM at each step. The likeness score of user u to frame k can be simply predicted by applying logistic function to the inner product between $\boldsymbol{p_u}$ and $\boldsymbol{q_k}$.

An alternative strategy is to only use the preference embedding as the "Zero State" of LSTM. However, we have empirically verified that this approach leads to unfavored performance for the task of personalized key frame recommendation.

Formally, suppose the time-synchronized comment of user u on frame k is tsc_{uk} with words $\boldsymbol{w_{tsc_{uk}}} = \{w_{tsc_{uk}}^0, w_{tsc_{uk}}^1, \dots w_{tsc_{uk}}^{s_{uk}-1}\}$, where s_{uk} is the length of the comment, and the size of the word vocabulary is defined as N^{word}. We formalize our architecture into an encoder-decoder framework similar to [6, 40].

More specifically, the user embedding and the frame embedding are first encoded into a joint preference embedding $\boldsymbol{e_{uk}^{pre_0}} = \boldsymbol{p_u} \odot \boldsymbol{q_k}$, where $\odot$ is element-wise multiplication. Then, given $\boldsymbol{e_{uk}^{pre_0}}$ and all the previously predicted words, the decoder predicts each word at iteration step t by a conditional distribution:

$$\boldsymbol{h_1} = LSTM(w_{tsc_{uk}}^0, \boldsymbol{e_{uk}^{pre_0}}) \tag{5}$$

$$\boldsymbol{h_t} = LSTM(\boldsymbol{h_{t-1}}, w_{tsc_{uk}}^{t-1}, \boldsymbol{e_{uk}^{pre_0}})\ t \in \{2, 3, \dots s_{uk}\} \tag{6}$$

$$p(w_{tsc_{uk}}^t | \boldsymbol{e_{uk}^{pre_0}}, \boldsymbol{w_{tsc_{uk}}^{0:t-1}}) = SOFTMAX(\boldsymbol{h_t}) \tag{7}$$

where $SOFTMAX()$ is an N^{word}-way softmax, $\boldsymbol{h_t}$ is the hidden state in LSTM at iteration t, $\boldsymbol{w_{tsc_{uk}}^{0:t-1}} = \{w_{tsc_{uk}}^{t-1}, w_{tsc_{uk}}^{t-2}, \dots, w_{tsc_{uk}}^0\}$ is the set of all previous words before iteration t, $LSTM()$ is the long short term memory (LSTM) net. At last, by simultaneously predicting users' likeness and time-synchronized comments, our final objective function to be maximized is:

$$
\begin{aligned}
L_2 = &\sum_{(u,k)\in O^+ \bigcup O^-} \sum_{t=1}^{s_{uk}-1} log\, p(w_{tsc_{uk}}^t | \boldsymbol{e_{uk}^{pre_0}}, \boldsymbol{w_{tsc_{uk}}^{0:t-1}}) \\
&+ \sum_{(u,k)\in O^+} log\, \hat{y}_{uk}^{TSC} + \sum_{(u,k)\in O^-} log\, (1 - \hat{y}_{uk}^{TSC})
\end{aligned}
\tag{8}
$$

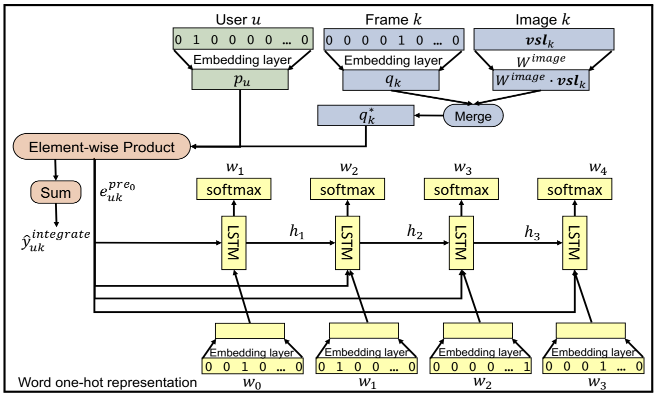

Figure 6: A model that directly combines the methods proposed in section 4.1 and 4.2. The output of the linear element-wise multiplication layer is directly used as an input of LSTM and to generate $\hat{y}_{uk}^{integrated}$.

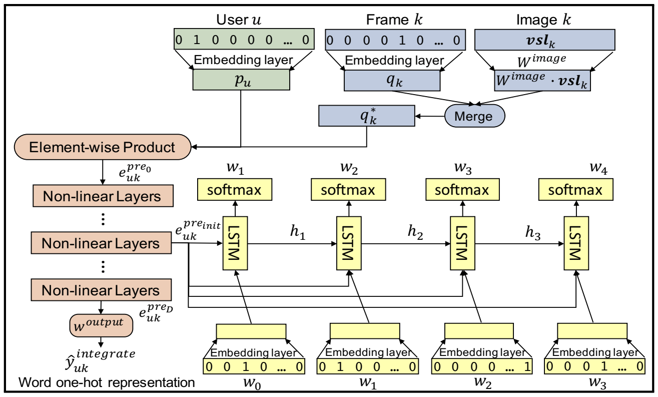

Figure 7: Our integrated recommender. D fully connected layers are introduced to capture the non-linear relationship between image and text features. Either the output of linear element-wise multiplication layer or the results from a non-linear layer can be used to initialize LSTM. $\hat{y}_{uk}^{integrated}$ is generated from the last non-linear layer.

where $\hat{y}_{uk}^{TSC} = LOGISTIC(\boldsymbol{p_u} \cdot \boldsymbol{q_k})$ is the prediction of user u's likeness score on frame k.

4.3 Integrated Recommender

In this subsection, we propose to jointly model frame image and time-synchronized comment in a unified framework.

Deep feature adapting. Intuitively, we may directly combine the above two models for user preference learning, which is shown in Figure 6. However, as image and text features come from quite different and heterogeneous information sources, the linear element-wise multiplication layer (see Figure 6) can be extremely biased when directly adapting such different information. To overcome this weakness, we stack several fully connected layers on top of the element-wise multiplication layer to capture the non-linear relationship among different features.

Formally, suppose the output of the element-wise multiplication layer is: $\boldsymbol{e_{uk}^{pre_0}}$, and there are totally D non-linear layers. Then the output of each non-linear layer and the final output can be derived as:

$$\boldsymbol{e_{uk}^{pre_0}} = \boldsymbol{p_u} \odot \boldsymbol{q_k^*} \tag{9}$$

$$\boldsymbol{e_{uk}^{pre_i}} = g^{nl}(\boldsymbol{W^i} \cdot \boldsymbol{e_{uk}^{pre_{i-1}}}) \; i \in \{1, 2, ...D\} \tag{10}$$

$$\hat{y}_{uk}^{integrated} = LOGISTIC(\boldsymbol{w^{output}} \cdot \boldsymbol{e_{uk}^{pre_D}}) \tag{11}$$

where g^{nl} is the active function, where we select Rectifier (ReLU) in our model because (1) it is practically more reasonable from a biological perspective [12], and (2) it can usually prevent deep models from over-fitting. $\boldsymbol{e_{uk}^{pre_i}}$ is the output of the i-th non-linear layer, $\boldsymbol{W^i}$ is the coefficient matrix used to weight $\boldsymbol{e_{uk}^{pre_{i-1}}}$, and $\boldsymbol{w^{output}}$ is a vector that maps $\boldsymbol{e_{uk}^{pre_D}}$ into a scalar so as to conduct logistic. Note that $\boldsymbol{w^{output}}$ is a parameter that needs to be learned by the model.

For now, we have described the key components (image modeling, text modeling and the D non-linear layers) of our final framework, and we further fuse them together (see Figure 7). Careful readers might have found that, except for the original preference embedding $\boldsymbol{e_{uk}^{pre_0}}$, any one of $\{\boldsymbol{e_{uk}^{pre_1}}, \boldsymbol{e_{uk}^{pre_2}}, ...\boldsymbol{e_{uk}^{pre_D}}\}$ can be used as the extra input of LSTM at each step, suppose we use $\boldsymbol{e_{uk}^{pre_{init}}}$ as the extra input, where $init \in \{0, 1, 2, ...D\}$ is a pre-defined constant. Our final framework can be learned by maximizing the following objective function:

$$L_3 = \alpha \sum_{(u,k) \in O^+ \bigcup O^-} \sum_{t=1}^{s_{uk}-1} log\, p\left(w_{tsc_{uk}}^t | \boldsymbol{e_{uk}^{pre_{init}}}, \boldsymbol{w_{tsc_{uk}}^{0:t-1}}\right) + (1-\alpha)\left(\sum_{(u,k) \in O^+} log\, \hat{y}_{uk}^{integrated} + \sum_{(u,k) \in O^-} log\left(1 - \hat{y}_{uk}^{integrated}\right)\right) \tag{12}$$

where α is a weighting parameter that balances the effects of different optimization objectives. Once the model has been learned, for a user u and a key frame k with visual feature $\boldsymbol{vsl_k}$, we can readily predict the likeness score of u to k by Equation (11), according to which we can further recommend u with the key frames that the user is most likely interested in.

5 EXPERIMENTS

In this section, we evaluate our proposed models focusing on the following three key research questions:

RQ 1: What is the performance of our final framework for the task of personalized key frame recommendation?
RQ 2: What are the effects of different types of information for personalized key frame recommendation?
RQ 3: Can the stacked non-linear layers promote the performance of personalized key frame recommendation?

We begin by introducing the experimental setup, and then report and analyze the experimental results to answer these research questions.

5.1 Experimental Setup

Dataset preprocess. The raw comments are generally pre-preprocessed (word segmentation and stop-word filtering) by an open-source Chinese natural language processing toolbox Jieba[7]. After that, we conduct more detailed pre-processing according to the special inherent characteristics of time-synchronized comments, and this is conducted on two aspects: on one hand, we manually remove the meaningless reviews, e.g., the ones at the beginning of the movies that are generally not relevant to the movie content; on the other hand, we map the slangs that express the same meaning (e.g., 2333..., namely, several 3's following a 2, which means "happiness" in online language environment) into a unified word (e.g., wonderful[8]) for more accurate modeling.

In our crawled dataset, the time stamp is recorded when a user sent an edited comment, however, the actual favored frame should be the one corresponding to the time when he/she began to type the comment, rather than the frame when the comment was posted out. As a result, we revise the time stamp by subtracting the time of typing according to the length of the comment and a person's general typewriting speed (approximately 40 words/minute). We pre-segment each movie as 1000 shots, and use the first frame as a shot's key frame. Because the frames in a shot are always very similar, all the commenting behaviors in a shot are seen as reviewing on its key frame, and we do not deliberately distinguish a shot from its key frame in the rest of the paper. To avoid "cold-start" problem, we remove those users with less than 100 time-synchronized comments, and finally sample a smaller dataset containing 40 users' 29,137 comments (20,312 positive ($pol_{uk} = 1$) and 8,825 negative ($pol_{uk} = 0$)) on 11,000 key frames.

Baselines. To demonstrate the effectiveness of our models, we adopt the following methods as baselines for performance comparison:

- **MostPopular:** This is a non-personalized static method utilizing user reviews, where for each user it just selects the most popularly positive key frames as the final results.
- **PMF:** The Probabilistic Matrix Factorization method proposed in [35], which is a frequently used state-of-the-art approach for rating-based optimization and prediction. We set the score of user u to key frame k as 1, if u commented on k with $pol_{uk} = 1$, and 0 otherwise.
- **BPR:** This is a well known ranking-based method [39] for user implicit feedback modeling, the preference pairs are constructed between the positively commented key frames and the other ones. In our experiments, we randomly sample one negative instance for each positive feedback.
- **HFT:** This is a stat-of-the-art method in terms of making rating prediction with textual reviews [31], as the rating information is absent in our dataset. We set the ratings of one's positively commented key frames as 1, and 0 otherwise for each user.
- **VBPR:** This is a stat-of-the-art visual-based recommendation method [15]. Similar to [15], the image features are pre-generated from the original key frame pictures using the Caffe deep learning framework [21].
- **KFRI:** This is a **K**ey **F**rame **R**ecommender based only on **I**mage features, which is proposed in section 4.1 with L_1 as its objective function.
- **KFRC:** This is a **K**ey **F**rame **R**ecommender based only on **C**omments, which is proposed in section 4.2 with L_2 as its objective function.

Evaluation method. If a user comments on a key frame with positive sentiment ($pol_{uk} = 1$), then this frame would be the one that attracts her, so the empirical experiments are conducted by comparing the predicted key frames with the true positive ones ($pol_{uk} = 1$). 30% of each user's positive key frames ($pol_{uk} = 1$) are selected as the test dataset, while the others are used for training. We adapt F_1-score and normalized discounted cumulative gain (NDCG) to evaluate the performance of the baselines and our proposed models.

Parameter settings. The hyper-parameters in our frameworks are tuned by conducting 5-fold cross validation, while the model parameters are first randomly initialized according to a uniform distribution in the range of (0, 1), and then updated by conducting stochastic gradient descent (SGD). The learning rate of SGD is determined by grid searching in the range of $\{1, 0.1, 0.01, 0.001, 0.0001\}$. We set the number of non-linear layers as $D = 3$, and to learn more abstractive features, their dimensions are empirically set as $\{40, 20, 10\}$ to form a tower structure [14].

We evaluate different number of latent factors K in the range of $\{50, 100, 150, 200, 250, 300\}$ for both user and frame vector representations. The number of negative samples K^{neg} is empirically set as 5, while the weighting parameter α is set as 0.5 to make different optimization parts equally contribute to the final results. For better performance, we leverage grid search technology to determine the batch size in the range of $\{64, 128, 256, 512, 1024\}$. When implementing the baselines, 5-fold cross validation and grid search technology are used to determine the parameters. Our experiments are conducted by predicting Top-5,10, and 20 favorite key frames respectively. All the models are repeated for 10 times, and we report the average as well as bound values as the final results for clear illustration.

5.2 Performance of Our Models (RQ1)

Different models (except **MostPopular**) may reach their best performance at various number of latent factors, so for each baseline, we implement it by setting the dimension as 50,100,150,200,250 and 300 respectively, and the best result is finally reported. From Figure 8, we can see: KFRCI achieves the best performance on both F_1 and $NDCG$ when recommending different number of key frames. It can on average enhance the performance by about 7.8% and 6.7% upon F_1-score and $NDCG$ respectively when compared with VBPR, which performs best among all the methods. Paired t-tests on the results also verify that the improvements are statistically significant

[7] https://github.com/fxsjy/jieba/tree/jieba3k

[8] Manually translated into English by the authors

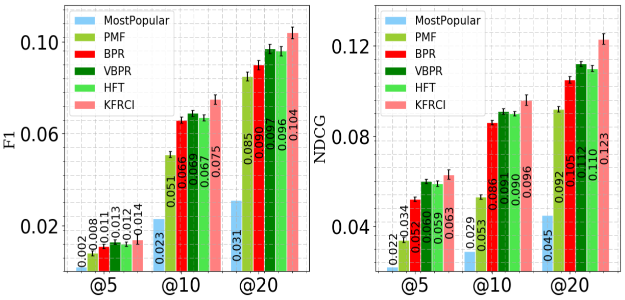

Figure 8: Comparison between our method and baseline methods. The values on the bar indicate the average performance of the corresponding models.

on 0.01 level. Among the baselines, PMF, as expected, performs better than MostPopular due to the consideration of diverse personalities. By directly optimizing the ranking objective, BPR shows better effectiveness compared with PMF on both F_1 and $NDCG$, which is consistent with the observations in [39]. By introducing textual or visual features, HFT/VBPR on average outperforms BPR by about 3.5%/5.4% on F_1 and 8.6%/6.1% on $NDCG$ respectively.

5.3 Effects of Different Information (RQ2)

For capturing more comprehensive user preference, frame images and time-synchronized comments are combined together in our final framework for joint modeling. However, different information sources may play different roles, in this section, we would like to study the influence of diverse information for our task of personalized key frame recommendation. To begin with, we compare our final framework KFRCI with KFRI and KFRC, which only use image or text information in their modeling processes. For fair comparison and to avoid disturbance of the deep architecture, we do not use any non-linear layers in KFRCI. The other parameters follow the above settings. From the results shown in Figure 10, we can see:

1. All the models can reach their best performance when the dimension falls in the range of [100, 150], while additional dimensions do not help promoting the performance. The reason may be that too many latent factors can lead to the over-fitting problem, which would weaken our models' generalization ability on the test dataset.

2. KFRI performs slightly better than KFRC in most cases, which tells us that, in our dataset, image features maybe more important compared with time-synchronized comments for the task of personalized key frame recommendation. This may be of the reason that although time-synchronized comments are helpful, they are too diverse and may include too much noise for capturing user inherent preference.

3. It is highly encouraging that although we did not use any non-linear layer, KFRCI exhibits higher performance compared with both KFRC and KFRI across all the dimensions. This observation demonstrates that the integration of visual and textual features can indeed help excavate more accurate user preference, which is in line with our intuition in Section 1.

Weighting parameter α. In this section, we study how the performance of KFRCI changes as the weighting parameter α increases from 0.1 to 0.9. In this experiment, the number of user/frame latent factors is fixed as 100, while the other parameters follow the settings in section 5.1. We predict Top-20 user favorite key frames, the results are shown in Figure 9, from which we can see: the performance (F_1@20) of KFRCI continues to rise until α reaches around 0.3, then after hovering approximately stable in the range of [0.3, 0.5], it begins to drop rapidly with the increase of α. This observation indicates that we should make a suitable balance between the two components in our final framework. Besides, similar results can be observed on $NDCG$@20.

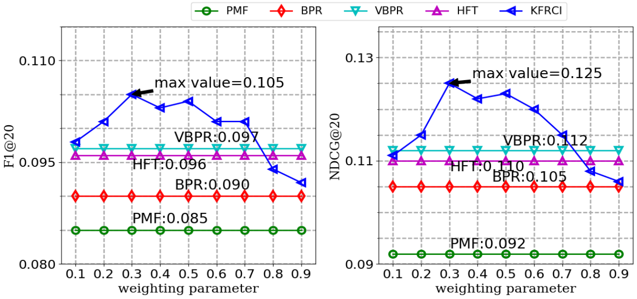

Figure 9: The influence of weighting parameter α. For clear comparison, we also list the performances of the other models, although they don't change with α. MostPopular is not listed herein because its performance is much lower.

5.4 Promotion of the Deep Architecture (RQ3)

In this section, we would like to test whether deep architectures are helpful to our task. To do so, we evaluate the performance of our final model KFRCI based on F_1 and $NDCG$ by changing the number of non-linear layers. Note that when there is no non-linear layer, we are actually evaluating the straightforward model as shown in Figure 6. In this experiment, the dimensions of the non-linear layers from the first layer to the output layer are set as $\{40, 20, 10, 5\}$, and the number of user/frame latent factors is fixed as 100. The output non-linear layer is used to link LSTM. All the other parameters follow the above settings.

The results are shown in Table 3, from which we can see that our model can reach its best performance when there are two or three non-linear layers, and introducing more non-linear layers does not bring positive effects. These observations indicate that deep model may be helpful for personalized key frame recommendation, however, only relatively small number of non-linear layers are required to capture the complex relationship among heterogeneous features.

"Extra input" of LSTM. When there are multiple non-linear layers, an obvious problem is that, which output should be selected as the "Extra input" of LSTM. So we further evaluate our model by using different layers' output $\boldsymbol{e}_{\boldsymbol{uk}}^{\boldsymbol{pre}_{\boldsymbol{init}}}$ as the LSTM "Extra input". Note that $init = 0$ means directly linking the output of element-wise

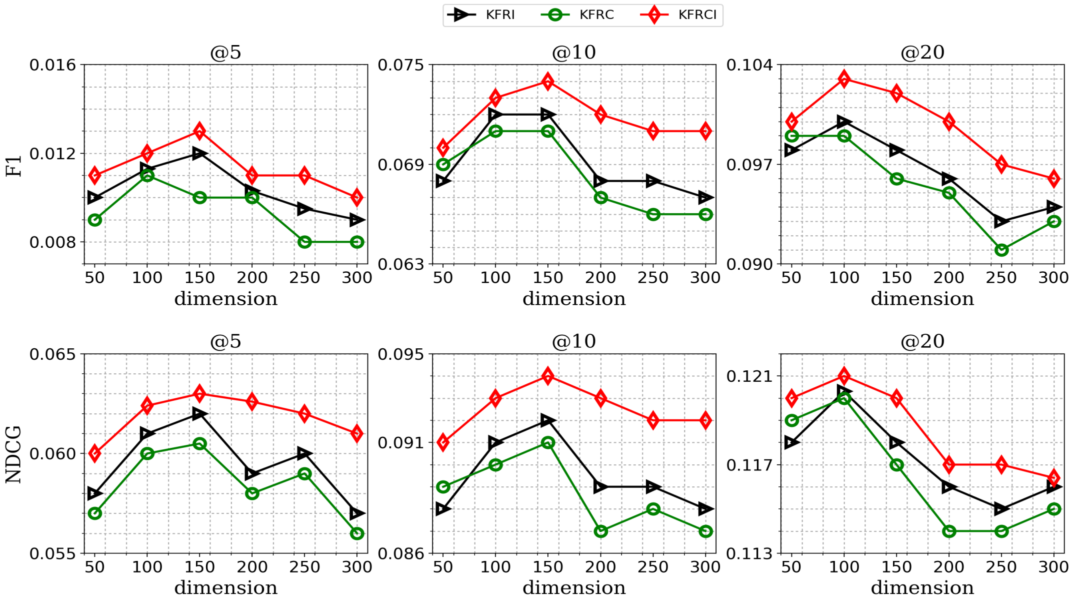

Figure 10: Evaluating the performances of our models when using different features. The dimension of the latent factors ranging from 50 to 300.

Table 3: The effect of deep architecture.

number of layers	0	1	2	3	4
F_1@5	0.012	0.014	**0.015**	0.014	0.013
NDCG@5	0.060	0.064	**0.065**	0.063	0.062
F_1@10	0.071	0.073	0.074	**0.075**	0.072
NDCG@10	0.092	0.094	0.095	**0.096**	0.091
F_1@20	0.103	0.104	**0.106**	0.104	0.103
NDCG@20	0.120	0.123	**0.124**	0.123	0.121

multiplication layer to the LSTM. In this experiment, we use 3 non-linear layers with the dimensions of {40, 20, 10}, other parameters follow the above settings.

From the results on F_1@20 and *NDCG*@20 shown in Table 4, we see that it can lead to improved performance when $init = 1, 2$ *or* 3 compared with $init = 0$, which manifests that introducing non-linear operations is important for better adapting the user/frame latent factors with the underlying motivations for generating time-synchronized comments.

Table 4: The effects of different LSTM “Extra inputs”.

init	0	1	2	3
F_1@20	0.103	0.104	**0.105**	0.104
NDCG@20	0.121	0.122	**0.124**	0.123

6 CONCLUSIONS AND OUTLOOK

In the paper, we propose the problem of personalized key frame recommendation for the first time. To do so, we propose to leverage the rich time-synchronized comment information in video sharing websites, and further design a novel framework that combines model-based collaborative filtering and long-short term memory network together to model user commented key frames and time-synchronized comments simultaneously. Comprehensive evaluation verified the effectiveness of our framework.

This is a first step towards our goal in personalized key frame recommendation, and there is much room for further improvements. For example, more other information (e.g. audio features) can be included to capture more comprehensive user preference, which may also bring us more inspiring insights on the inherent natures of the user preference patterns upon video key frames. Beyond personalized key frame recommendation, our work also points to promising future directions in personalized video summarization, personalized image captioning, and personalized story telling based on images or videos.

ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1160894, NSFC 61602176 and NSFC 61672231. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Y. Bao, H. Fang, and J. Zhang. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *AAAI*, pages 2–8, 2014.

[3] Y. Cao, J. Li, X. Guo, S. Bai, H. Ji, and J. Tang. Name list only? target entity disambiguation in short texts. In *EMNLP*, pages 654–664, 2015.

[4] X. Chen, Z. Qin, Y. Zhang, and T. Xu. Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th international ACM SIGIR conference on Research & development in information retrieval.* ACM, 2016.

[5] X. Chen, P. Wang, Z. Qin, and Y. Zhang. Hlbpr: A hybrid local bayesian personal ranking method. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 21–22. International World Wide Web Conferences Steering Committee, 2016.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.

[8] Q. Cui, S. Wu, Q. Liu, and L. Wang. A visual and textual recurrent neural network for sequential prediction. *arXiv preprint arXiv:1611.06668*, 2016.

[9] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM, 2014.

[10] N. Ejaz, T. B. Tariq, and S. W. Baik. Adaptive key frame extraction for video summarization using an aggregation mechanism. *Journal of Visual Communication and Image Representation*, 23(7):1031–1040, 2012.

[11] G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, volume 9, pages 1–6. Citeseer, 2009.

[12] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.

[13] Y. Gong and X. Liu. Video summarization using singular value decomposition. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 174–180. IEEE, 2000.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[15] R. He and J. McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1510.01784*, 2015.

[16] X. He, T. Chen, M.-Y. Kan, and X. Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670. ACM, 2015.

[17] X. He, M. Gao, M.-Y. Kan, and D. Wang. Birank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):57–71, 2017.

[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference Companion on World Wide Web (WWW)*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[19] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 549–558. ACM, 2016.

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[22] R. M. Jiang, A. H. Sadka, and D. Crookes. Advances in video summarization and skimming. In *Recent Advances in Multimedia Signal Processing and Communications*, pages 27–50. Springer, 2009.

[23] W. Jiang, C. Cotton, and A. C. Loui. Automatic consumer video summarization by audio and visual analysis. In *ICME*, pages 1–6, 2011.

[24] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, pages 2698–2705, 2013.

[25] G. Kim, L. Sigal, and E. P. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014.

[26] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, pages 1346–1353, 2012.

[27] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu. Video summarization via transferrable structured learning. In *WWW*, pages 287–296. ACM, 2011.

[28] Y. Li, T. Zhang, and D. Tretter. An overview of video abstraction techniques. Technical report, Technical Report HPL-2001-191, HP Laboratory, 2001.

[29] H. Liu, J. He, T. Wang, W. Song, and X. Du. Combining user preferences and user opinions for accurate recommendation. *Electronic Commerce Research and Applications*, 12(1):14–23, 2013.

[30] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang. A generic framework of user attention model and its application in video summarization. *IEEE transactions on multimedia*, 7(5):907–919, 2005.

[31] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[32] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

[33] E. Mendi and C. Bayrak. Shot boundary detection and key frame extraction using salient region detection and structural similarity. In *Proceedings of the 48th Annual Southeast Regional Conference*, pages 66–67, 2010.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[35] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[36] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. *Journal of Information Processing*, 15(2):316, 1992.

[37] J. Peng, Y. Zhai, and J. Qiu. Learning latent factor from review text and rating for recommendation. In *2015 7th International Conference on Modelling, Identification and Control (ICMIC)*, pages 1–6. IEEE, 2015.

[38] Z. Ren, S. Liang, P. Li, S. Wang, and M. de Rijke. Social collaborative viewpoint regression with explainable recommendations. *Under submission*, page 10, 2016.

[39] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[40] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[41] Y. Tan, M. Zhang, Y. Liu, and S. Ma. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *IJCAI*. AAAI, 2016.

[42] D. Tang, B. Qin, and T. Liu. Learning semantic representations of users and products for document level sentiment classification. In *ACL*, pages 1014–1023, 2015.

[43] D. Tang, B. Qin, T. Liu, and Y. Yang. User modeling with neural network for review rating prediction. In *IJCAI*, pages 1340–1346, 2015.

[44] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[45] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *SIGKDD*, pages 1235–1244, 2015.

[46] H. Wang, S. Xingjian, and D.-Y. Yeung. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In *NIPS*, pages 415–423, 2016.

[47] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua. Event driven web video summarization by tag localization and key-shot identification. *Multimedia, IEEE Transactions on*, 14(4):975–985, 2012.

[48] Z. Wang, M. Kumar, J. Luo, and B. Li. Sequence-kernel based sparse representation for amateur video summarization. In *Proceedings of the 2011 joint ACM workshop on Modeling and representing events*, pages 31–36, 2011.

[49] B. Wu, E. Zhong, B. Tan, A. Horner, and Q. Yang. Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In *SIGKDD*, pages 721–730, 2014.

[50] C.-Y. Wu, A. Beutel, A. Ahmed, and A. J. Smola. Explaining reviews and ratings with paco: Poisson additive co-clustering. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 127–128. International World Wide Web Conferences Steering Committee, 2016.

[51] Y. Wu and M. Ester. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *WSDM*, pages 199–208. ACM, 2015.

[52] Y. Xian, J. Li, C. Zhang, and Z. Liao. Video highlight shot extraction with time-sync comment. In *Workshop on Hot Topics in Planet-scale mobile computing and online Social networking*, pages 31–36, 2015.

[53] H. Xu, Y. Zhen, and H. Zha. Trailer generation via a point process-based visual attractiveness model. In *IJCAI*, pages 2198–2204, 2015.

[54] J. You, G. Liu, L. Sun, and H. Li. A multiple visual models based perceptive analysis framework for multilevel video summarization. *CSVT, IEEE Transactions on*, 17(3):273–285, 2007.

[55] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[56] W. Zhang, Q. Yuan, J. Han, and J. Wang. Collaborative multi-level embedding learning from reviews for rating prediction. In *IJCAI*. ACM, 2016.

[57] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM, 2014.

[58] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 866–870. IEEE, 1998.