Personalized Key Frame Recommendation

Xu Chen[1], Yongfeng Zhang[2], Qingyao Ai[2], Hongteng Xu[3], Junchi Yan[4], Zheng Qin[1]
[1]School of Software, Tsinghua University, Beijing, 10084, China
[2]College of Information and Computer Science, University of Massachusetts Amherst, MA 01003
[3]School of Electrical and Computer Engineering, Georgia Institute of Technology, GA 30332
[4]East China Normal University, IBM Research - China, Shanghai, China
{xu-ch14,qinzh}@mails.tsinghua.edu.cn,{yongfeng,aiqy}@cs.umass.edu,hxu42@gatech.edu,jcyan@sei.ecnu.edu.cn

ABSTRACT

Video key frame extraction has long been a research task of fundamental importance in a variety of applications, such as online movie preview, content summarization, and video information retrieval. Although the related techniques have been largely investigated in the research community, current approaches of key frame extraction mainly base themselves on image-only features, and fall into the non-personalized manners without the consideration of per-user interests. However, in a real-world scenario, different users may cast different interests on the style or content of video images, and thus they may be attracted by different key frames even for the same video, which makes key frame extraction an inherently personalized process.

In this paper, to bridge the above gap, we propose and investigate personalized key frame recommendation. To do so, we design a novel collaborative neural recommender to model key frame images as well as time-synchronized comments simultaneously. By user personalization based on her/his previously reviewed frames and the posted comments, we are able to profile different user interests in a unified multi-modal space, and can thus provide 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.

CCS CONCEPTS

•Online Information Services →*Web-based Services;* **•Information Search and Retrieval** →*Information Filtering;*

KEYWORDS

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

ACM Reference format:
Xu Chen[1], Yongfeng Zhang[2], Qingyao Ai[2], Hongteng Xu[3], Junchi Yan[4], Zheng Qin[1]. 2017. Personalized Key Frame Recommendation. In *Proceedings of The 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, Aug 2017 (SIGIR'17),* 11 pages.
DOI: 10.475/123_4

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGIR'17, Tokyo, Japan
© 2017 Copyright held by the owner/author(s). 123-4567-24-567/08/06…$15.00
DOI: 10.475/123_4

1 INTRODUCTION

Videos have been an important information source ever since their existence in many online applications, such as video sharing websites, digital video broadcast, etc. However, the management of such unstructured data is very challenging due to its huge volume and high complication. To operate the video data more efficiently, key frame extraction methods [7, 16, 49, 52] are proposed to capture the major elements in a video in terms of content. With the help of the extracted key frames, web users can readily understand the general style or the content of a movie even without watching the whole video [31, 37].

Despite the many encouraging and emerging improvements in the field of key frame extraction, existing methods only focus on how to precisely extract the key frames to summarize the video, but fail to further distinguish key frames' attractiveness for different users. However, in practice, people may have diverse personality and individual likes or dislikes, and thus they may be drawn by various key frames even for the same video. Without the tailored key frames, people are likely to miss their favorite videos due to the misleading non-personalized key frames. Therefore, in real applications, an important question should be 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] sheds 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 [41], 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-sync comment "*… I like his overcoat, it looks cool and also must be very comfortable with good quality*". From the content of the time-sync comment we can understand the particular aspects that attract the user's attention, such as, clothing quality and comfort level, while the frame image can further acquaints us with the visual features that she is interested in, such as clothing style, texture, etc, which are usually difficult to be described with

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

text. As a result, we believe heterogenous information sources like text and image can be complementary with each other in terms of user profiling, and may help to promote personalized key frame recommendation when being integrated in proper manners. By leveraging all the historical implicit feedback as well as the features (image and text) of users' interest, we can collaboratively match a target customer with her potentially favorite frames.

Based on the above motivation and intuition, we describe and analyze a **C**ollaborative **N**eural **R**ecommender in this paper to make personalized key frame recommendation by modeling the **T**extual features collected from user time-sync comments and the **V**isual features extracted from frame images simultaneously (called **CNRTV**). The main building block of our proposed method is to integrate the power of model-based collaborative filetering and long-short term memory network. The carefully designed collaborative filtering component aims to capture personalized user preferences based on image features, while the modified long-short term memory network component aims to model user time-sync 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 multimodal space so as to facilitate comprehensive user profiling and accurate key frame recommendation.

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

- We propose personalized key frame recommendation, and to the best of our knowledge, this is the first work to investigate key frame extraction in a personalized manner.
- To solve the above novel problem, we present a novel hybrid neural architecture to model user time-sync comments and frame image features simultaneously.
- We perform extensive experiments to verify the superiority of our proposed model for the task of personalized key frame recommendation.

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 methods with experimental results. Conclusions and outlooks of this work are presented in section 6.

2 RELATED WORK

2.1 Time-sync Comments

Time-Synchronized Comment (TSC) is first introduced in [41] for automatic video shot tagging. In this work, the authors propose a novel method to extract time-sync video tags by automatically exploiting crowdsourcing comments. [44] 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.

2.2 Key Frame Extraction

Key frame extraction, or called video summarization, has attracted much research interest and many works have been proposed in the past. As aforementioned, most of existing works are based on image processing and computer vision techniques. Early works [10, 22] extract visual features of frames and cluster frames accordingly. To improve the performance, other side information beyond visual features is considered in recent work, including the viewer attention [24, 46, 47], audio signal [17], subtitles [21], etc. Moreover, semantic information has also been exploited to summarize videos, including special events [39, 40], key people and objects [18, 20], and storylines [19]. However, the methods above neither consider the problem of personalized key frame recommendation nor take advantage of the time-sync comments in their algorithmic frameworks.

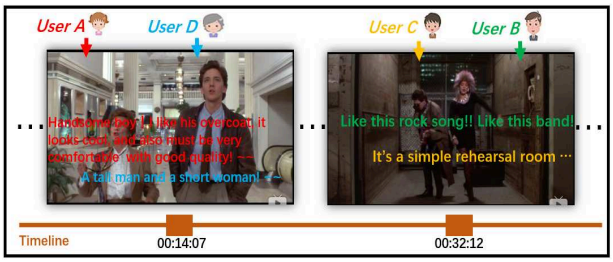


Figure 1: A simple example of TSC. Different users may express real-time opinions directly upon their interested frames. The comments are manually translated into English by the authors.

2.3 Review-based Recommendation

Incorporating user reviews into traditional recommendation algorithms has attracted much research interest and many models have been proposed in the past [2, 3, 6, 8, 13, 23, 25, 33, 36, 42, 43, 50, 51]. According to the method that textual reviews are processed, 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, [25, 36] link the latent factors in rating data with the topics in the textual review to generate more accurate recommendations, and [6, 43] 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, [50] combines traditional matrix factorization technology with word2vec [29] 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 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, [51] uses multi-matrix factorization to generate explainable recommendations based on the extracted product features. [3] further captures user interested product features in a learning to rank manner.

Our model partly falls into the first category, however, compared with the aforementioned methods, we can capture word sequential orders mirrored in time-sync comments, which has been ignored in previous works.

2.4 Image-based Recommendation

Recently, there is a trend to incorporate visual features into the research of personalized recommendation [5, 12, 27]. Specifically, [12] proposes to incorporate visual signals into matrix factorization method to enhance the performance of recommendation and to alleviate the cold start problem. [5] further adopts product images as well as item category and title information to make sequential predictions. [27] leverages visual features to find visually complementary items to a query image.

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-sync comments, which is another main difference when compared with our model.

3 PRELIMINARY AND PROBLEM DEFINITION

3.1 Dataset Inspection

We crawled all the time-synchronized comments till December 10th, 2015 from the category of movie on Bilibili[4], which – commonly known as the B-site – is a well-known time-synchronized comments video website. 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)	1133750
Total number of movies (#movies)	7166
Total number of TSCs (#TSCs)	11842166
Ave. TSCs per movie	1652.5
Ave. users per movie	465.9
Ave. TSCs per user	10.4
Max/Min number of TSCs for a movie	8028/101
Max/Min number of users for a movie	3370/1
Max/Min number of TSCs for a user	68236/1

As can be seen in Figure 2, the quantity of active users is relatively small, and most users only send 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

In this section, we formally introduce the problem definition. Suppose there are N users $\boldsymbol{u} = \{u_1, u_2, ..., u_N\}$ and M movies/videos $\boldsymbol{v} = \{v_1, v_2, ..., v_M\}$. For each movie $v_i \in \boldsymbol{v}$, we pre-segment it into K shots $\boldsymbol{s}^{v_i} = \{s_1^{v_i}, s_2^{v_i}, ...s_K^{v_i}\}$, and generate the key frame for each shot using some non-personalized key frame extraction method such as [28, 32]. The key frame in shot $s_j^{v_i}$ is defined as $f_j^{v_i}$.

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

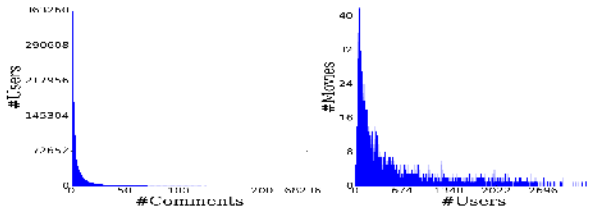


Figure 2: On the left is the relation between the number of comments and users, while on the right is the relation between the number of users and movies.

Let user $u's$ time-sync comment on key frame f be c_{uf} with words $\boldsymbol{w}^{c_{uf}} = \{w_0^{c_{uf}}, w_1^{c_{uf}}, ...w_{l_{uf}-1}^{c_{uf}}\}$, where l_{uf} is the length of the comment. The visual feature of f is defined as $\boldsymbol{d}_f^{image}$, and $\boldsymbol{F}$ represents the whole set of key frames among all the videos.

Given users' historical time-sync comments $\boldsymbol{W} = \{\boldsymbol{w}^{c_{uf}} | u \in \boldsymbol{u}, f \in \boldsymbol{F}\}$ as well as the visual features $\boldsymbol{D} = \{\boldsymbol{d}_f^{image} | f \in \boldsymbol{F}\}$, for a target user u and one of her unseen movie $v_i \in V$ with pre-selected key frames $\{f_1^{v_i}, f_2^{v_i}, ...f_K^{v_i}\}$, our task is to find a function g to re-rank these key frames according to $u's$ interest, that is, $g(\{f_1^{v_i}, f_2^{v_i}, ...f_K^{v_i}\} | u, \boldsymbol{W}, \boldsymbol{D}) = \{f_{o1}^{v_i}, f_{o2}^{v_i}, ...f_{oK}^{v_i}\}$, where $\{o1, o2, ...oK\}$ is an ordering of $\{1, 2, ...K\}$. 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 COLLABORATIVE NEURAL 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 the personalized information. Lastly, we further design a unified framework to jointly model frame images as well as time-synchronized comments, in which visual and textual features can mutually promote each other for the task of personalized key frame recommendation.

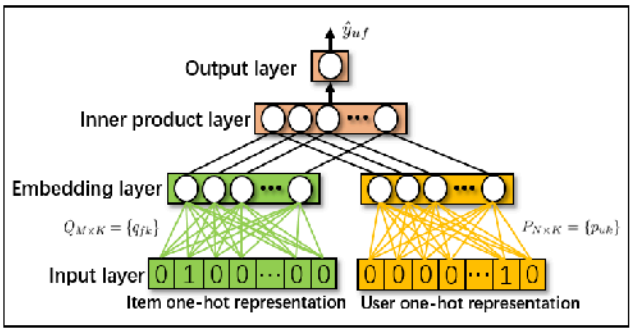


Figure 3: MF as neural network.

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 $\{v_1, v_2, \ldots, v_M\}$.
$\boldsymbol{s^{v_i}}$	The set of K shots $\{s_1^{v_i}, s_2^{v_i}, \ldots s_K^{v_i}\}$ for movie $\boldsymbol{v_i}$.
$f, f_j^{v_i}, F$	An arbitrary key frame,the key frame of shot $s_j^{v_i}$ and the set of all key frames.
$\boldsymbol{d_f^{image}}, \boldsymbol{D}$	The preprocessed visual feature of frame f, the set of all visual features.
$\boldsymbol{w^{c_{uf}}}, \boldsymbol{W}$	The word list $\{w_0^{c_{uf}}, w_1^{c_{uf}}, \ldots w_{l_{uf}-1}^{c_{uf}}\}$ in $u's$ time-sync comment on key frame f, the set of all time-sync comments.
$\boldsymbol{p_u}, \boldsymbol{q_f}$	Latent factors of user u and frame f.
O^+, O^-	The set of positive, sampled negative feedback.
K^{neg}	Number of negative instances.
N^{word}	Size of the word vocabulary.
$\boldsymbol{h_t}$	The hidden state in LSTM at iteration t
L	The number of non-linear layers.
$\boldsymbol{e^{pre_0}}, \boldsymbol{e^{pre_1}}, \ldots \boldsymbol{e^{pre_L}}$	Preference embeddings.
$\boldsymbol{W^{image}}, \boldsymbol{W^i}, \boldsymbol{w^{output}}$	Weighting matrix that maps $\boldsymbol{d_f^{image}}$ into a K dimensional vector, the coefficient matrix used to weight $\boldsymbol{e^{pre_{i-1}}}$, a vector that maps $\boldsymbol{e^{pre_L}}$ into a scalar.
$g^{merge}, g^{logistic}, g^{LSTM}$	The merge function, logistic function, LSTM network.
$\hat{y}_{uf}^{image}, \hat{y}_{uf}^{TSC}, \hat{y}_{uf}^{integrated}$	Predicted $u's$ likeness to f when using image/TSC/integrated information.

For the clarity and integrality, we first re-describe the widely used matrix factorization (MF) model as a neural network. Formally, let $\boldsymbol{p_u}$ and $\boldsymbol{q_f}$ represent the latent factors of user u and item f, then the likeness (or score) of u to f is usually predicted as $\hat{y}_{uf} = \boldsymbol{p_u^T q_f}$. In the context of neural network (see Figure 3), the user/item ids with one-hot format can be seen as inputs feeding 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 $\hat{y}_{uf}$ is computed as the vector inner product between $\boldsymbol{p_u}$ and $\boldsymbol{q_f}$.

4.1 Image Feature Modeling

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.

It is well known to all that convolutional neural network (CNN) is a powerful tool to process images. However, to make our model more scalable and practically available, similar to [12], rather than re-training the whole convolutional neural network, we chose to use a pre-trained Caffe deep learning framework [15] to generate visual features from raw images, which would greatly speed up our optimization process. Our particular choice of CNN is 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 f, we use the output of FC7, namely, the second fully-connected layer, as the final visual feature $\boldsymbol{d_f^{image}}$, which is a feature vector of length 4096.

Suppose the dimension of user/frame latent factors (embedding) is K, and again, let $\boldsymbol{p_u}$ and $\boldsymbol{q_f}$ be the latent factors of user u and frame f respectively. Intuitively, image features should serve as an important role for reflecting frame characters. When it comes to our model,we should let $\boldsymbol{d_f^{image}}$ directly influence the final embedding of frame f. We first project $\boldsymbol{d_f^{image}}$ into a K dimensional vector (denoted as the purple vectors in the figure) for space unifying. And then we explicitly merge this derived vector with the original frame latent factors $\boldsymbol{q_f}$ to generate $f's$ final embedding (blue vector). Lastly, the user latent factors together with the newly generated embedding are fed into the inner product layer to compute the final prediction.

Let $\boldsymbol{W^{image}} \in R^{K \times 4096}$ be the weighting matrix that maps $\boldsymbol{d_f^{image}}$ into a K dimensional vector, then the likeness of user u to frame f, $\hat{y}_{uf}^{image} \in [0, 1]$, can finally be predicted by:

$$\hat{y}_{uf}^{image} = g^{logistic}\Big(\boldsymbol{p_u} \cdot g^{merge}(\boldsymbol{q_f}, \boldsymbol{W^{image}} \boldsymbol{d_f^{image}})\Big). \quad (1)$$

where $g^{logistic}(x) = \frac{1}{1+e^{-x}}$ is the logistic function, "$\cdot$" is the inner product, $g^{merge} : R^K \times R^K \rightarrow R^K$ is a function that merges two K dimension vectors into one. The particular choice of g^{merge} in our model is a simple element-wise multiplication, i.e.,

$$g^{merge}\Big((a_1, a_2, \ldots a_K), (b_1, b_2, \ldots b_K)\Big) = (a_1 b_1, a_2 b_2, \ldots a_K b_K) \quad (2)$$

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

In our framework, we use the binary cross-entropy as our loss function to model the implicit feedback, whose superiority has been explored and demonstrated in [45]; based on this, our final objective function to be maximized is:

$$\begin{aligned} L_1 &= log \prod_{(u,f)} (\hat{y}_{uf}^{image})^{y_{uf}} (1 - \hat{y}_{uf}^{image})^{(1-y_{uf})} \\ &= log \prod_{(u,f) \in O^+} \hat{y}_{uf}^{image} \prod_{(u,f) \in O^-} (1 - \hat{y}_{uf}^{image}). \\ &= \sum_{(u,f) \in O^+} log\, \hat{y}_{uf}^{image} + \sum_{(u,f) \in O^-} log\, (1 - \hat{y}_{uf}^{image}). \end{aligned} \quad (3)$$

where y_{uf} is the ground truth that would be 1 if u has commented on f, and 0 otherwise. O^+ is the set of positive feedbacks, which is $O^+ : \{(u, f) | u \textit{ has commented on } f\}$, while O^- is the set of sampled negative feedback, namely, $O^- \subseteq \{(u, f) | u \textit{ did not comment on } f\}$.

In the training phase, for a user u and one of her interacted key frame f, we uniformly sample K^{neg} negative instances in the same video where f lies in, and the parameters can be learned via stochastic gradient descent (SGD).

4.2 Text Feature Modeling

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

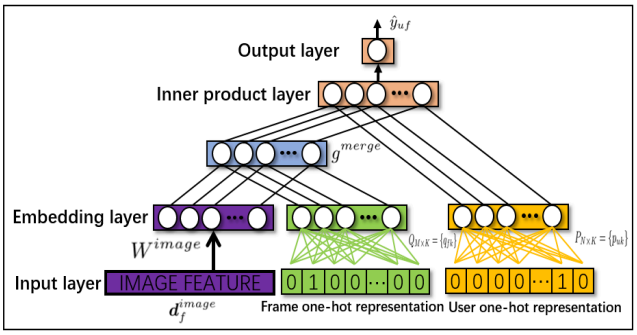


Figure 4: The framework of image feature modeling. The preprocessed image feature is merged with frame latent factors to derive a new embedding, which is then multiplied by the user latent factors to generate the prediction.

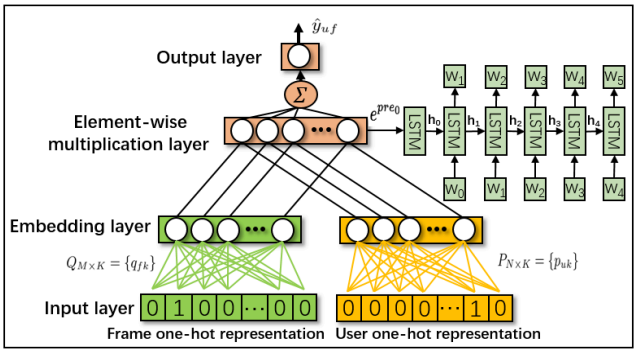


Figure 5: The framework of text feature modeling. The preference embedding e^{pre_0} is severed as the "Zero State" of a LSTM network, which is further used for generating time-sync comment. The likeness of user u to frame f can be simply predicted by conducting logistic function on the inner product between p_u and q_f.

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) [14] network, which has been successfully applied to a number of sequence modeling tasks such as machine translation [1], image caption [38], and video classification [48].

Intuitively, the content of a time-sync 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 shown in Figure 5, we first merge the user and frame latent factors into a preference embedding, and then feed this embedding as the "Zero State" to initialize the LSTM network, which further generates the time-sync comment word by word.

An alternative strategy to fuse the preference embeddings into LSTM is to feed it as an extra input to LSTM at each step. However, we have empirically verified that this approach leads to unfavored performance for the task of personalized key frame recommendation, which is in line with the findings in [38].

Formally, suppose the time-sync comment of user u on frame f is c_{uf} with words $\boldsymbol{w}^{c_{uf}} = \{w_0^{c_{uf}}, w_1^{c_{uf}}, ...w_{l_{uf}-1}^{c_{uf}}\}$, where l_{uf} 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 [4, 35].

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

$$p(w_t^{c_{uf}} | \boldsymbol{e^{pre_0}}, \boldsymbol{w}_{0:t-1}^{c_{uf}}) = g^s(\boldsymbol{h_t}, w_{t-1}^{c_{uf}}, \boldsymbol{e^{pre_0}}) \quad (4)$$

$$\boldsymbol{h_t} = g^{LSTM}(\boldsymbol{h_{t-1}}, w_{t-1}^{c_{uf}}, \boldsymbol{e^{pre_0}}) \quad (5)$$

where g^s is an N^{word}-way softmax, $\boldsymbol{h_t}$ is the hidden state in LSTM at iteration t, $\boldsymbol{w}_{0:t-1}^{c_{uf}} = \{w_{t-1}^{c_{uf}}, w_{t-2}^{c_{uf}}, ...w_0^{c_{uf}}\}$ is the set of all previous words before iteration t, g^{LSTM} is the long short term memory (LSTM) net. At last, by simultaneously predicting users' likeness and time-sync comments, our final objective function to be maximized is:

$$L_2 = \sum_{(u,f)\in O^+ \bigcup O^-} \sum_{t=1}^{l_{uf}-1} log\, p(w_t^{c_{uf}} | \boldsymbol{e^{pre_0}}, \boldsymbol{w}_{0:t-1}^{c_{uf}}) + \sum_{(u,f)\in O^+} log\, \hat{y}_{uf}^{TSC} + \sum_{(u,f)\in O^-} log\,(1 - \hat{y}_{uf}^{TSC}) \quad (6)$$

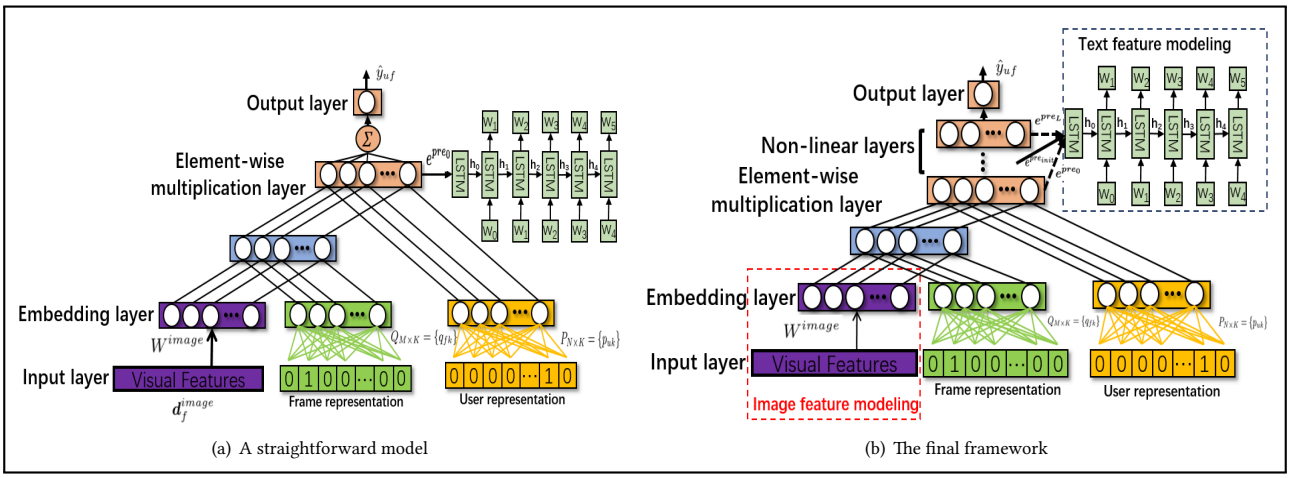
where $\hat{y}_{uf}^{TSC} = g^{logistic}(\boldsymbol{p_u} \cdot \boldsymbol{q_f})$ is the prediction of u's likeness on f. In the training phase, the length of comment in a negative instance is set as 0 to represent that there is actually no comment, and all the parameters can also be learned by conducting stochastic gradient descent (SGD).

4.3 Integration of the Image and Text Features

As discussed before, image and text features can uncover user preference from different aspects, and they may complementarily help each other to boost the performance of personalized key frame recommendation. In this subsection, we propose to jointly model frame image and time-sync comment in a unified framework. Intuitively, we may directly combine the above two models for user preference learning, which is shown in Figure 6(a).

However, as image and text features come from quite different and heterogenous information sources, the linear element-wise multiplication layer (see Figure 6(a)) can be extremely biased when directly adapting very different information. To overcome this weakness, we stack several fully connected layers on top of element-wise multiplication layer to capture the non-linear relationship among different features.

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



(a) A straightforward model (b) The final framework

Figure 6: (a) 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 to initialize LSTM and generate $\hat{y}_{uf}^{integrated}$. (b) Our final framework. L 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. $\hat{y}_{uf}^{integrated}$ is generated from the last non-linear layer.

as:

$$\boldsymbol{e}^{\boldsymbol{pre}_0} = \boldsymbol{p}_{\boldsymbol{u}} \odot g^{merge}(\boldsymbol{q}_f, \boldsymbol{W}^{\boldsymbol{image}} \boldsymbol{d}_f^{image}) \tag{7}$$

$$\boldsymbol{e}^{\boldsymbol{pre}_i} = g^{nl}(\boldsymbol{W}^{\boldsymbol{i}} \cdot \boldsymbol{e}^{\boldsymbol{pre}_{i-1}}) \; i \in \{1, 2, ...L\} \tag{8}$$

$$\hat{y}_{uf}^{integrated} = g^{logistic}(\boldsymbol{w}^{\boldsymbol{output}} \cdot \boldsymbol{e}^{\boldsymbol{pre}_L}) \tag{9}$$

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 [9], and (2) it can usually prevent deep models from overfitting. $\boldsymbol{e}^{\boldsymbol{pre}_i}$ is the output of the i-th non-linear layer, $\boldsymbol{W}^{\boldsymbol{i}}$ is the coefficient matrix used to weight $\boldsymbol{e}^{\boldsymbol{pre}_{i-1}}$, and $\boldsymbol{w}^{\boldsymbol{output}}$ is a vector that maps $\boldsymbol{e}^{\boldsymbol{pre}_L}$ into a scalar so as to conduct logistic. $\odot$, g^{merge}, $\boldsymbol{W}^{\boldsymbol{image}}$ and $\boldsymbol{d}_f^{image}$ are the same as defined in section 4.1.

For now, we have described the key components (image modeling, text modeling and the L non-linear layers) of our final framework, and we further fuse them together (see Figure 6(b)). Careful readers might have found that, besides the output of element-wise multiplication layer $\boldsymbol{e}^{\boldsymbol{pre}_0}$, each of $\{\boldsymbol{e}^{\boldsymbol{pre}_1}, \boldsymbol{e}^{\boldsymbol{pre}_2}, ...\boldsymbol{e}^{\boldsymbol{pre}_L}\}$ can also be used to initialize the LSTM network, and our following experiments have also verified that different LSTM "Zero States" can indeed lead to different performances. Suppose, we use $\boldsymbol{e}^{\boldsymbol{pre}_{init}}$ as the "Zero State" of LSTM, where $init \in \{0, 1, 2, ...L\}$ is a pre-defined constant. Our final framework can be learned by maximizing the following objective function:

$$L_3 = \alpha \sum_{(u,f)\in O^+ \bigcup O^-} \sum_{t=1}^{l_{uf}-1} log\, p\left(w_t^{c_{uf}} | \boldsymbol{e}^{\boldsymbol{pre}_{init}}, \boldsymbol{w}_{0:t-1}^{c_{uf}}\}\right) + (1-\alpha)\left(\sum_{(u,f)\in O^+} log\, \hat{y}_{uf}^{integrated} + \sum_{(u,f)\in O^-} log\left(1 - \hat{y}_{uf}^{integrated}\right)\right) \tag{10}$$

where α is a weighting parameter that balances the effects of different optimization objects. Once the model has been learned, for a user u and a key frame f with visual feature $\boldsymbol{d}_f^{\boldsymbol{image}}$, we can readily predict the likeness score of u to f by equation 9, 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 by 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 kinds 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 first pre-processed for word segmentation and stop-word filtering by an open-source natural language processing toolbox Jieba[5]. After that, we conduct finer-grained processing of the time-sync comments on two aspects: on one hand, we remove the reviews at the beginning of the movies that are generally not relevant to the movie content, and 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[6]) for more accurate modeling.

In our crawled dataset, the time stamp is recorded when a user sent an edited comment, however, the actually favoured frame is at the moment he/she began to type the comment, rather than the time 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 the key frame of a shot. Because the frames in a shot are always very similar that focus on the same scene, 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 and better examine our models' collaborative filtering capability, we remove those users with less than 100 time-sync comments, and finally sample a smaller dataset[7] containing 40 users' 29173 reviews on 11000 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 commented key frames as the final results.
- **PMF:** The Probabilistic Matrix Factorization method proposed in [30], which is a frequently used stat-of-the-art approach for rating-based optimization and prediction. We set score of user u to key frame f as 1, if u commented on f, and 0 otherwise.
- **BPR:** This is a well known ranking-based method [34] for user implicit feedback modeling. Preference pairs are constructed between the commented key frames and the uncommented 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 recommendation based on textual reviews [26]. To construct the (implicit) rating matrix, we set the rating of a user's commented key frame as 1, and 0 otherwise.
- **VBPR:** This is a stat-of-the-art visual-based recommendation method [12]. Similar to [12], the image features are pre-generated from the original key frame pictures using the Caffe deep learning framework [15].
- **CNRV:** This is a **c**ollaborative **n**eural **r**ecommender based only on **v**isual features (CNRV), which is proposed in section 4.1 with L_1 as its objective function.
- **CNRT:** This is a **c**ollaborative **n**eural **r**ecommender based only on **t**extual features (CNRT), which is proposed in section 4.2 with L_2 as its objective function.

Evaluation method. We assume that users' commented key frames are those that attracted them, so the empirical experiments are conducted by comparing the predicted key frames with the actually commented key frames of the corresponding users, and 30% of each user's commented key frames 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. Our models are implemented based on TensorFlow[8]. 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 3, and to learn more abstractive features, the dimensions are empirically set as $\{40, 20, 10\}$ to form a tower structure [11]. The word embedding size in LSTM is fixed as 256, which is a common setting in the field of natural language processing. We evaluate different number of frame/user latent factors K in the range of $\{50, 100, 150, 200, 250, 300\}$. The number of negative samples K^{neg} is empirically set as 5, while the weighting parameter α is set as 0.5 to let 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 both the average as well as the bound values for clear illustration.

5.2 Performance of Our Models (RQ1)

Different models (except **MostPopular**) may reach their best performance at different number of latent factors. For each baseline, we first implement it by setting the dimension as 25,50,100,150,200,250 and 300 respectively, and then we report the best result. From Table 3, we can see: CNRTV 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 16.9% and 11.3% upon F_1-score and $NDCG$ respectively, as compared with VBPR, which performs the best among all the methods. Paired t-tests on

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

[6] Manually translated into English by the authors

[7] We will make the whole as well as the sampled dataset publicly available to the research community.

[8] Our source codes are available at [anonymized website]

Table 3: Comparison between our method and baseline methods. The blacked values indicate significant improvements against the best baseline on 0.01 level.

		MostPopular	PMF	BPR	VBPR	HFT	CNRTV($K = 100$)
F_1	@5	$0.002^{+0.000}_{-0.000}$	$0.008^{+0.001}_{-0.001}$	$0.011^{+0.001}_{-0.001}$	$0.013^{+0.002}_{-0.001}$	$0.012^{+0.002}_{-0.001}$	$\mathbf{0.018^{+0.001}_{-0.001}}$
	@10	$0.023^{+0.000}_{-0.000}$	$0.051^{+0.001}_{-0.001}$	$0.066^{+0.001}_{-0.001}$	$0.069^{+0.001}_{-0.001}$	$0.067^{+0.002}_{-0.002}$	$\mathbf{0.079^{+0.001}_{-0.001}}$
	@20	$0.031^{+0.000}_{-0.000}$	$0.085^{+0.001}_{-0.001}$	$0.090^{+0.001}_{-0.001}$	$0.097^{+0.001}_{-0.001}$	$0.096^{+0.001}_{-0.001}$	$\mathbf{0.109^{+0.001}_{-0.001}}$
	Ave.	$0.019^{+0.000}_{-0.000}$	$0.048^{+0.001}_{-0.001}$	$0.056^{+0.001}_{-0.001}$	$0.059^{+0.001}_{-0.001}$	$0.058^{+0.002}_{-0.001}$	$\mathbf{0.069^{+0.001}_{-0.001}}$
$NDCG$	@5	$0.022^{+0.000}_{-0.000}$	$0.034^{+0.001}_{-0.001}$	$0.052^{+0.001}_{-0.001}$	$0.060^{+0.002}_{-0.001}$	$0.059^{+0.001}_{-0.001}$	$\mathbf{0.066^{+0.001}_{-0.001}}$
	@10	$0.029^{+0.000}_{-0.000}$	$0.053^{+0.001}_{-0.001}$	$0.086^{+0.002}_{-0.001}$	$0.091^{+0.001}_{-0.002}$	$0.090^{+0.001}_{-0.002}$	$\mathbf{0.099^{+0.001}_{-0.001}}$
	@20	$0.045^{+0.000}_{-0.000}$	$0.092^{+0.001}_{-0.001}$	$0.105^{+0.001}_{-0.002}$	$0.112^{+0.001}_{-0.001}$	$0.110^{+0.001}_{-0.002}$	$\mathbf{0.130^{+0.001}_{-0.001}}$
	Ave.	$0.023^{+0.000}_{-0.000}$	$0.059^{+0.001}_{-0.001}$	$0.081^{+0.001}_{-0.001}$	$0.088^{+0.001}_{-0.001}$	$0.086^{+0.001}_{-0.002}$	$\mathbf{0.098^{+0.001}_{-0.001}}$

the results also verify that the improvements are statistically significant 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 higher effectiveness compared with PMF on both F_1 and $NDCG$, which is consistent with the observations in [34]. 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-sync comments are combined together in our final framework for joint modeling. However, different information sources may play different roles. In this section, we'd like to study the influence of diverse information for our task of personalized key frame recommendation. To begin with, we compare our final framework CNRTV with CNRT and CNTV, which only use image or text information in their modeling processes. For fair comparison and avoiding disturbance of the deep architecture, we did not use any non-linear layers in CNRTV. The other parameters follow the above settings. From the results shown in Figure 8, 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 overfitting, which weakens the generalization ability of our models on the test dataset.

2. CNRV performs slightly better than CNRT in most cases, which implies that in our dataset, image features may be more important compared with time-sync comments on the task of personalized key frame recommendation. The reason may be that, although time-sync comments are helpful, they can be semantically diverse and may include noise to capture the frame-level user preference.

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

Weighting parameter α. We have compared our final framework with its two simplified versions, for further investigating the effects of different information, we are also curious about what roles do the different features play in our final framework itself. In this section, we study how the performance of CNRTV 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 7, from which we can see: the performance (F_1@20) of CNRTV 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, in our final framework, the modeling of user implicit feedback based on frame images may play a more important role as compared with the time-sync comment modeling. Besides, similar results can be seen on $NDCG$@20.

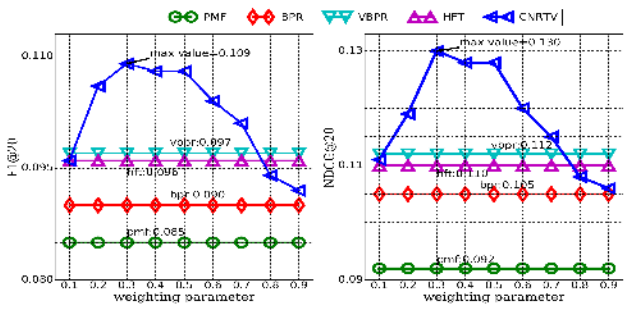


Figure 7: The influence of weighting parameter α. For clear comparison, we also list the performances of the other models, although they do not change with α. MostPopular is not listed here because its performance is on a much lower scale.

5.4 Promotion of the Deep Architecture (RQ3)

In this section, we would like to test whether deep architecture helps in our task. To do so, we evaluate the performance of our final model CNRTV 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(a). In this experiment, the dimensions of the non-linear layers from down to top are set as $\{40, 20, 10, 5\}$, the number of user/frame latent factors is fixed as 100. The output of the top non-linear layer is used to link LSTM. All the other parameters follow the above settings. The results are shown in Table 4, 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 gives positive effects. These observations indicate that deep models can be helpful for personalized key frame recommendation, however, only a relatively small number of non-linear layers are required to capture the complex relationship among heterogeneous features.

Table 4: The effect of deep architecture.

number of layers	0	1	2	3	4
F_1@5	0.014	0.016	**0.019**	0.018	0.016
$NDCG$@5	0.063	0.065	**0.068**	0.066	0.064
F_1@10	0.073	0.076	0.078	**0.079**	0.078
$NDCG$@10	0.093	0.096	0.098	**0.099**	0.097
F_1@20	0.103	0.108	**0.110**	0.109	0.108
$NDCG$@20	0.123	0.129	**0.131**	0.130	0.131

"Zero State" of LSTM. When there are multiple non-linear layers, an obvious problem is to decide which output should be selected as the "Zero State" of LSTM. As a result, we further evaluate our model by using the output of different layers as the "Zero State" e^{init} of LSTM. Note that $init = 0$ means directly linking the output of element-wise multiplication layer to the LSTM (see the dashed lines in Figure 6(b)). In this experiment, we use 3 non-linear layers with the dimensions of $\{40, 20, 10\}$, and other parameters follow the above settings. From the results on F_1@20 and $NDCG$@20 shown

Table 5: The effects of different LSTM "Zero State".

$init$	0	1	2	3
F_1@20	0.105	0.108	**0.109**	0.108
$NDCG$@20	0.124	0.129	**0.131**	0.130

in Table 5, we can see that it can lead to improved performance when $init = 1, 2$ *or* 3, which manifests that introducing non-linear operations is important to better adapt the user/frame latent factors with the underlying motivations for generating time-sync comments (meaning of the sentence is not very clear).

6 CONCLUSION 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-sync comment information in video sharing websites, and further design a novel framework that integrates the power of model-based collaborative filtering and long-short term memory network to model user commented key frames and time-sync comments simultaneously. Experimental results on three key research questions verified the effectiveness of our framework on different aspects.

This is a first step towards our goal in personalized key frame recommendation, and there is much room for further improvements. For example, we can consider more multimode information (e.g. audio features) to capture user preferences more comprehensively, which may also give us inspiring insights on the nature of user preference patterns on video key frames, and consider other non-uniform sampling methods to train our model according to the specific characteristics of different personalized key frame recommendation tasks. In addition, the proposed collaborative neural recommender based on visual and textual features is a general framework, and it is a possible direction to extend this framework into other tasks such as personalized product recommendation in e-commerce.

ACKNOWLEDGEMENT

We thank the reviewers for the careful reviews and constructive suggestions. This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1160894. 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] 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.
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.
[10] Y. Gong and X. Liu. Video summarization using singular value decomposition. In *CVPR*, volume 2, pages 174–180, 2000.
[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
[12] R. He and J. McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1510.01784*, 2015.
[13] 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*

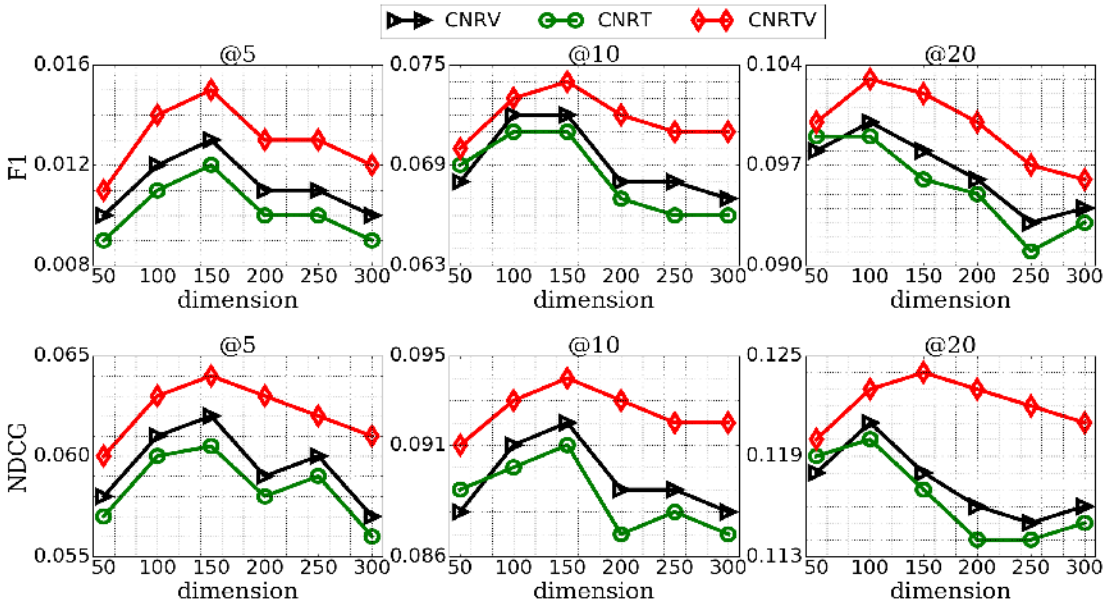


Figure 8: Evaluating the performances of our models when using different features. The dimension of the latent factors ranging from 50 to 300.

on Conference on Information and Knowledge Management, pages 1661–1670. ACM, 2015.

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

[15] 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.

[16] 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.

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

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

[19] 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.

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

[21] 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.

[22] Y. Li, T. Zhang, and D. Tretter. An overview of video abstraction techniques. *HP Laboratories Palo Alto*, 2001.

[23] 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.

[24] 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. *Multimedia, IEEE Transactions on*, 7(5):907–919, 2005.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

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

[31] A. G. Money and H. Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121–143, 2008.

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

[33] 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.

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

[35] 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.

[36] 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.

[37] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 3(1):3, 2007.

[38] 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.

[39] 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.

[40] 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.

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

[42] 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.

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

[44] Y. Xian, J. Li, C. Zhang, and Z. Liao. Video highlight shot extraction with time-sync comment. In *Proceedings of the 7th International Workshop on Hot Topics in Planet-scale mObile computing and online Social neTworking*, pages 31–36, 2015.
[45] H. Z. L. N. X. H. T.-S. C. Xiangnan He, Lizi Liao. Neural collaborative filtering. 2017.
[46] H. Xu, Y. Zhen, and H. Zha. Trailer generation via a point process-based visual attractiveness model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2198–2204. AAAI Press, 2015.
[47] 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.
[48] 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.
[49] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern recognition*, 30(4):643–658, 1997.
[50] W. Zhang, Q. Yuan, J. Han, and J. Wang. Collaborative multi-level embedding learning from reviews for rating prediction.
[51] 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.
[52] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *ICIP*, volume 1, pages 866–870. IEEE, 1998.