Interpretable Recommendation via Attraction Modeling: Learning Multilevel Attractiveness over Multimodal Movie Contents

Liang Hu[1,2], Songlei Jian[1,3], Longbing Cao[1], Qingkui Chen[2]
[1]Advanced Analytics Institute, University of Technology, Sydney
[2]Institute of Network Computing & IoT, University of Shanghai for Science and Technology
[3]College of Computer, National University of Defense Technology, China
rainmilk@gmail.com, jiansonglei@163.com, longbing.cao@uts.edu.au, chenqingkui@usst.edu.cn

Abstract

New contents like blogs and online videos are produced in every second in the new media age. We argue that attraction is one of the decisive factors for user selection of new contents. However, collaborative filtering cannot work without user feedback; and the existing content-based recommender systems are ineligible to capture and interpret the attractive points on new contents. Accordingly, we propose attraction modeling to learn and interpret user attractiveness. Specially, we build a multilevel attraction model (MLAM) over the content features - the story (textual data) and cast members (categorical data) of movies. In particular, we design multilevel personal filters to calculate users' attractiveness on words, sentences and cast members at different levels. The experimental results show the superiority of MLAM over the state-of-the-art methods. In addition, a case study is provided to demonstrate the interpretability of MLAM by visualizing user attractiveness on a movie.

1 Introduction

Nowadays, new media, e.g., social websites and online video stream, dominates the traffic of Internet. Huge amount of new contents including news, blogs and videos, are produced in every second. As a result, a lot of content-sharing platforms have emerged in recent years. For example, Twitter is an online news and social networking service where users post and interact with messages. Users will be recommended the latest news that most potentially attract them when they login. By 2010, Netflix's streaming business has become the largest source of Internet streaming traffic in North America in the evening. As reported, "About 75 percent to 80 percent of what people watch on Netflix comes from what Netflix recommends, not from what people search for" [Timothy, 2013]. Obviously, recommendation becomes the first choice for users to consume contents in the new media age.

Classical recommender systems (RS) may help users to find interesting contents according to similar users' history with collaborative filtering (CF) technique. However, CF cannot work in the cold-start cases, e.g., a news article has not been rated by any users. Content-based filtering (CBF) [de Gemmis *et al.*, 2015] finds contents by the semantic similarity so it does not suffer from the above issue. Intuitively, the user selection of contents is often determined by some attractive points, e.g., a place in a news, or an actor of a movie. However, current CBF approaches cannot capture the attractive points leading to user selection. In recent years, more and more researchers argue that only focusing on improving the accuracy may hurt RS [McNee *et al.*, 2006; Hu *et al.*, 2017b]. Following this argument, instead of aiming at higher recommendation accuracy, we pay more attention to finding and interpreting the attractive points in available contents, although our model can still achieve comparable accuracy performance.

Specially, we aim to model user attractiveness over contents to interpret user selection by assuming that attraction is one of the strongest motivations to make the final decision. For example, a person may choose a movie due to the attraction on a movie star ignoring other factors of that movie. Similarly, we may like a song due to one or two heart-touching lyrics even though we cannot remember the whole song. Obviously, these attractive points instead of the whole movie or song result in the selection. Furthermore, attraction is a subjective feeling which is different from person to person. For example, a person may like a movie as s/he is attracted by an actress while another person selects this movie as attracted by the director. Therefore, modeling attraction is a critical task to interpret user behavior, which is not limited to interpret user selection in RSs. Attention mechanism can assign focus on selective parts where a related context is given. It has been shown effective in various tasks such as machine translation [Bahdanau *et al.*, 2014] and image captioning [Vinyals *et al.*, 2015]. Obviously, attention mechanism shares some common ideas with attraction modeling. The main difference is that attention mechanism focuses on salient parts of an object in an objective way without considering user difference whereas attraction modeling aims to find personal focus on a content in a subjective way. In this work, we incorporate user context into attention mechanism for modeling subjective attraction.

Since the internet video stream has accounted for the major traffic in this new media age, in this paper, we take online movies as the representative case to study attraction modeling. In particular, the story and the cast members, e.g., actors, directors and writers, are two most important aspects of a movie to attract audience. One one hand, when a person reads

the story of a movie, s/he may be caught by some attractive words in a sentence, e.g., a character's name. Moreover, only a few sentences of the core plot instead of all sentences may actually attract user's attention. Accordingly, we build a multilevel neural model on the story (textual content) to capture word-level, sentence-level, and story-level attraction. On the other hand, cast members (categorical content) of a movie are another important factor to attract users so we build another neural model to weight the attraction over each cast member and generate a representation of the whole cast. At the top level, we create a joint representation of story and cast representation to score attractiveness. Due to the complementation of story (textual data) and cast members (categorical data), we build a multimodal neural model to integrate the information from both types of data to comprehensively capture user attraction.

The contributions of this paper are summarized as follows:

- We model human attraction on new contents, using movie content as the study case, to capture and interpret the motivation on user selections.
- A multimodal neural network model with subjective attention mechanism is designed to learn the multilevel personal attraction on the story and the attraction on the cast members of a movie.
- Extensive experiments on a real-world dataset are conducted to evaluate the above design. All results show that our approach can achieve comparable performance with the state-of-the-art methods. Moreover, we demonstrate the statistical user attractiveness on movies to interpret the recommendation results.

2 Related Work

Briefly, we present the two most relevant aspects to our work: (1) the content-based recommender systems, and (2) the attention mechanism on contents.

CF is not capable of recommending new content because there is no other users' feedback on this new item. In comparison, the CBF approach can recommend the latest contents, but it may lead to overspecialization when a user is associated with very limited contents [Balabanović and Shoham, 1997]. Kompan et al. [2010] proposed a content-based news recommendation method, where users are assigned into a cluster of similar news articles according to their browse history. Since each user is assigned to a news cluster, this method lacks of personalization for each user and fails to model the diversity of user's tastes across multiple news clusters. To take advantage of both CF and CBF, some hybrid recommendation methods were proposed. fLDA [Agarwal and Chen, 2010] generalized the supervised topic model (sLDA) [Mcauliffe and Blei, 2008] by using the latent topics learning from textual content for recommendation. Similar to fLDA, Wang et al. [2011] proposed a hybrid method for scientific article recommendation by combining matrix factorization and LDA. Musto et al. [Musto *et al.*, 2016] proposed to learn word embedding from Wikipedia, and represent user profile as the centroid of the embedding vectors of the items the user previously liked. The goal of these methods is recommendation other than interpreting user attraction as in this work.

Attention mechanism has been shown effective in various tasks such as machine translation [Bahdanau *et al.*, 2014] and image captioning [Vinyals *et al.*, 2015]. Recently, some researchers have employed attention mechanism to model textual content. Yang et al. [2016] proposed hierarchical attention networks for document classification, where attention mechanisms are respectively applied at the word and sentence level, enabling it to attend to more or less important content when constructing the document representation. Denil et al. [2014] use convolutional neural networks (CNN) to transform word embedding in each sentence into the embedding for the entire sentence. At the document level, another CNN is used to transform sentence embedding into a document embedding vector. These methods try to find salient words or sentences from documents without considering user factors. To relieve the workload of editors for selecting articles, Wang et al. [2017] proposed a dynamic attention deep model (DADM) to recommend articles, where each article is represented by a vector using character-level text modeling [Kim *et al.*, 2016]. However, these attentive words and sentences do not mean the attractive points to all users since each user may have quite different attention. Moreover, the content of an item, e.g., a movie, often consists of multiple types of data, not limited to text. Therefore, we design a multilevel attraction neural network to model personal attention on multimodal contents.

3 Problem Statement

We denote the movie set as $\mathcal{M} = \{m_1, \cdots, m_N\}$. For each movie $m \in \mathcal{M}$, it consists of a textual story, $\mathcal{S}_t$, and a set of cast members $\mathcal{C}_m = \{c_1, \cdots, c_{N_m}\}$. For each story $\mathcal{S}_t$, it consists of N_t sentences, $\mathcal{S}_t = \{s_1, \cdots, s_{N_t}\}$. For each sentence $s \in \mathcal{S}_t$, it consists of a set of N_s words, $\{w_{s,1}, \cdots, w_{s,N_s}\}$. We denote the user set as $\mathcal{U}$ to model their attraction on movies. Given a user $u \in \mathcal{U}$, her user profile about previously liked movies is denoted $\mathcal{M}_u = \{m_{u,1}, \cdots, m_{u,N_u}\}$.

Given a movie $m \in \mathcal{M}_u$, one of our tasks is to learn the attractiveness over words for each sentence, and the attractiveness over sentences of the story $\mathcal{S}_t$ from user u's perspective and generate story-level representation $\mathbf{h}^t$. Another task is to weight the attractiveness on the cast members $\mathcal{C}_m$ and generate attraction-based cast-level representation $\mathbf{h}^c$. Then, we can use $\mathbf{h}^t$ and $\mathbf{h}^c$ to score the attractiveness on movie m. When the parameters of attraction model are learned, we can compute personal attractiveness scores over a set of candidate movies $\mathcal{M}_C$ for recommendation and interpret the recommendation with the highlighted sentences of the story and which actors may attract the target user.

4 Multilevel Attraction Model

The overview of the architecture of our model is illustrated in Figure 1. This model consists of three main parts: cast attraction module (left), user module (middle) and story attraction module (right). In the story module, we build a multilevel attraction model to score attractiveness over words and sentences for each user. Similarly, we build another attraction model to score attractiveness over cast members in the

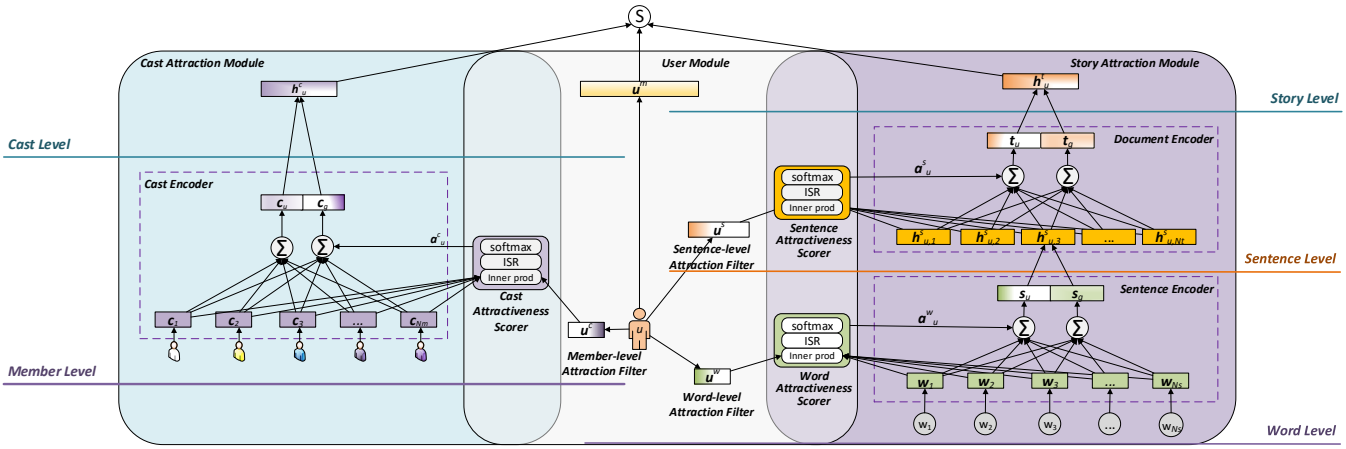

Figure 1: The architecture of multilevel attraction model over movies with two modalities: Cast (left) and Story (right)

cast module. Finally, we compute the personal attractiveness score of a movie on the top of these modules using the cast-level embedding and the story-level embedding.

Different from classical attention mechanism without considering subjective user personalization, we design a bottom-up multilevel attraction model over the text of a movie story to learn personally attractive embeddings. At the word level, our model scores the attractiveness over words in a sentence, and then encodes a sentence embedding based on the word embedding vectors weighted by their attractiveness scores. Recursively, our model encodes the story embedding based on the sentence embedding vectors weighted by their attractiveness. In the following subsections, we give more technical details about these modules.

4.1 Multilevel Story Attraction Module

Sentence Encoder with Word Attraction Filters

Given a set of words $\{w_1, \cdots, w_{N_s}\}$ of a sentence s in story $\mathcal{S}_t$ of movie $m_{u,i}$ liked by user u, we aim to score the attractiveness over words from u's perspective. First, we map each word w_i into a word embedding vector $\mathbf{w}_i \in \mathbb{R}^L$, where L denotes the length of the embedding vector. Then, we input these word embedding vectors $\{\mathbf{w}_1, \cdots, \mathbf{w}_{N_s}\}$ to the word attractiveness scoring module (illustrated in the overlapped part of user module and story attraction module). We use $\mathbf{u}_u^w \in \mathbb{R}^L$ to denote the word-level attraction filter of user u. Then, we compute the attractiveness score $a_{u,i}^w$ in terms of the inner product between $\mathbf{u}_u^w$ and each word embedding $\mathbf{w}_i$

$$a_{u,i}^w = \mathbf{u}_u^{w\top} \mathbf{w}_i, \quad i \in \{1, \cdots, N_s\} \tag{1}$$

The normalized weight can be computed by $softmax(a_i) = e^{a_i} / \sum_j e^{a_j}$. However, $a_{u,i}^w$ in Eq. 1 can be arbitrarily large, which makes softmax to output a weight close to 1 on the maximum $a_{u,i}^w$. Furthermore, a large a_i easily makes the exponential function overflow in implementation. To resolve this problem, we impose an inverse squared root function isr^α to bound the value of a_i.

$$isr^\alpha(x) = \frac{x}{\sqrt{1 + \alpha x^2}}, \quad isr \in (-\alpha^{-\frac{1}{2}}, \alpha^{-\frac{1}{2}}) \tag{2}$$

We can use the parameter α to control the upper bound and lower bound. A large α makes the upper bound and lower bound close to 0; as a result, the softmax tends to output uniform weights. A small $\alpha = 0.001$ has the range $(-31.6, 31.6)$ which guarantees the exponential function not overflow and the softmax tends to output a single large weight. In practice, α needs to be tuned with data. Accordingly, the normalized attractiveness score on word w_i is

$$\bar{a}_{u,i}^w = softmax(isr^{\alpha=4}(a_{u,i}^w)) \tag{3}$$

We find that $\alpha = 4$ performs good through our experiments.

Since $\bar{a}_{u,i}^w$ scores user attractiveness on each word, we create a personal attraction sentence embedding $\mathbf{s}_u$ by weighted sum over word embedding vectors using $\bar{a}_{u,i}^w$.

$$\mathbf{s}_u = \sum_i \bar{a}_{u,i}^w \mathbf{w}_i \tag{4}$$

$\mathbf{s}_u$ largely encodes the information from the most attractive words and discard the information from unattractive ones. To measure the overall attractiveness of the sentence s at the sentence level, we need to preserve the major information of a sentence apart from the most attractive part encoded by $\mathbf{s}_u$. Therefore, we use another *attraction-free* filter $\mathbf{g}^w$ to extract the major information over all words.

$$\bar{a}_i^w = softmax[isr^{\alpha=32}(\mathbf{g}^w \mathbf{w}_i + b^w)], i \in \{1, \cdots, N_s\} \tag{5}$$

The filter weights are computed similar to user attractiveness (cf. Eq. 3), where the major difference is that the weight vector $\mathbf{g}$ and the bias b^w are user independent and α of isr is set to 32 to keep relatively uniform information for all words. As a result, the *attraction-free* sentence embedding $\mathbf{s}_g$ is encoded:

$$\mathbf{s}_g = \sum_i \bar{a}_i^w \mathbf{w}_i \tag{6}$$

Then, we concatenate the attraction-encoded sentence embedding and the attraction-free sentence embedding, $[\mathbf{s}_u, \mathbf{s}_g]$, and input it into the $\tanh$ neural network layer to jointly encode $\mathbf{s}_u$ and $\mathbf{s}_g$ into a comprehensive sentence encoding $\mathbf{h}_u^s$ with the parameters $\mathbf{W}^s$ and $\mathbf{b}^s$

$$\mathbf{h}_u^s = \tanh([\mathbf{s}_u, \mathbf{s}_g]\mathbf{W}^s + \mathbf{b}^s) \tag{7}$$

Story Encoder with Sentence Attraction Filters

Once we obtain the sentence embeddings (cf. Eq. 7) for all sentences of the story $\mathcal{S}_t$ from the sentence encoder as presented above, we can build a story encoder over these sentence embedding vectors at the sentence level. As shown in the right-hand of Figure 1, the structure of story encoder is very similar to the sentence encoder. Therefore, we briefly introduce this story encoder in this subsection.

Given the sentence-level user attraction filter, $\mathbf{u}_u^s$, of user u, and the sentence embedding vectors $\{\mathbf{h}_{u,1}^s, \cdots, \mathbf{h}_{u,N_t}^s\}$, the attractiveness scores over $\mathbf{u}_u^s$ and each $\mathbf{h}_{u,i}^s$ is given by

$$a_{u,i}^s = \mathbf{u}_u^{s\top} \mathbf{h}_{u,i}^s, \quad i \in \{1, \cdots, N_t\} \tag{8}$$

Accordingly, we can obtain the normalized attractiveness score $\bar{a}_{u,i}^s$ on each sentence embedding and the corresponding attraction-based story embedding $\mathbf{t}_u$:

$$\bar{a}_{u,i}^s = softmax(isr^{\alpha=2}(a_{u,i}^s)) \tag{9}$$

$$\mathbf{t}_u = \sum_i \bar{a}_{u,i}^s \mathbf{h}_{u,i}^s \tag{10}$$

where $\alpha = 2$ is set throughh experiments. Apart from encoding the most attractive sentences, we use the attraction-free story embedding to preserve other information of the story for the follow-up movie-level attraction scoring.

$$\bar{a}_i^s = softmax[isr^{\alpha=16}(\mathbf{g}^s \mathbf{h}_{u,i}^s + b^s)], i \in \{1, \cdots, N_t\} \tag{11}$$

$$\mathbf{t}_g = \sum_i \bar{a}_i^s \mathbf{h}_{u,i}^s \tag{12}$$

Then, the comprehensive story embedding $\mathbf{h}_u^t$ is encoded with the parameters $\mathbf{W}^t$ and $\mathbf{b}^t$:

$$\mathbf{h}_u^t = \tanh([\mathbf{t}_u, \mathbf{t}_g]\mathbf{W}^t + \mathbf{b}^t) \tag{13}$$

4.2 Multilevel Cast Attraction Module

The architecture of cast attraction module is similar to the story attraction module. First, we map the cast members $\{c_1, \cdots, c_{N_m}\}$ of the movie $m_{u,i}$ into embedding vectors $\{\mathbf{c}_1, \cdots, \mathbf{c}_m\}$. Given the user attraction filter $\mathbf{u}_u^c$, the attractiveness score over each cast members is:

$$a_{u,i}^c = \mathbf{u}_u^{c\top} \mathbf{c}_i, \quad i \in \{1, \cdots, N_m\} \tag{14}$$

Accordingly, the normalized attractiveness score $\bar{a}_{u,i}^c$ and the attraction-based cast embedding $\mathbf{c}_u$ w.r.t. user u are given:

$$\bar{a}_{u,i}^c = softmax(isr^{\alpha=1}(a_{u,i}^c)) \tag{15}$$

$$\mathbf{c}_u = \sum_i \bar{a}_{u,i}^c \mathbf{c}_i \tag{16}$$

For the follow-up movie-level attraction scoring, we need to preserve the overall attraction-free cast information besides the attractive cast member embedding as done in the story attraction module.

$$\bar{a}_i^c = softmax[isr^{\alpha=16}(\mathbf{g}^c \mathbf{c}_i + b^c)], i \in \{1, \cdots, N_m\} \tag{17}$$

$$\mathbf{c}_g = \sum_i \bar{a}_i^c \mathbf{c}_i \tag{18}$$

Finally, we obtain the comprehensive cast embedding $\mathbf{h}_u^c$, which is encoded with the parameters $\mathbf{W}^c$ and $\mathbf{b}^c$

$$\mathbf{h}_u^c = \tanh([\mathbf{c}_u, \mathbf{c}_g]\mathbf{W}^c + \mathbf{b}^c) \tag{19}$$

Algorithm 1 The learning procedure for a mini-batch

1: $\mathcal{B} \leftarrow$ GetMinibatch($\{\mathcal{M}_u\}$) from all user-movie pairs
2: $\mathcal{N} \leftarrow$ Sample contractive movies $m_{u,j}$ for each $m_{u,i} \in \mathcal{B}$
3: Compute mini-batch loss using Eq. 21:
$L_{\mathcal{B}} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{\langle m_{u,i}, m_{u,j} \rangle \in \langle \mathcal{B}, \mathcal{N} \rangle} L_{m_{u,i} \succeq m_{u,j}}$
4: Update parameters: $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \Gamma_{Adam}(\nabla_{\Theta} L_{\mathcal{B}})$

4.3 Optimization Objective and Training

Movie Attraction Scoring

After we obtain the comprehensive story embedding $\mathbf{h}_u^t$ from story attraction module and the comprehensive cast embedding $\mathbf{h}_u^c$ from cast attraction module. We concatenate them as the joint multimodal movie embedding $[\mathbf{h}_u^t, \mathbf{h}_u^c]$. Then, the attraction scores on the movie m can be computed with the user's movie-level filter $\mathbf{u}_u^m$ over $[\mathbf{h}_u^t, \mathbf{h}_u^c]$:

$$S_{m_u} = \mathbf{u}_u^{m\top} [\mathbf{h}_u^t, \mathbf{h}_u^c] \tag{20}$$

Ranking Loss

In real-world scenarios, explicit like/dislike data are often not available; instead, data like watch records and click logs are much more easily obtained. In such cases, we only have one-class data [Hu *et al.*, 2016; 2017a] which cannot be directly used to differentiate user preferences. Learning from one-class preference data is often treated as a ranking problem [Rendle *et al.*, 2009]. Given a user u, we can construct a contrastive pair to specify the attractiveness order, that is, we have the order $m_{u,i} \succeq m_{u,j}$ over a movie ($m_{u,i} \in \mathcal{M}_u$) explicitly selected by u and an unselected movie ($m_{u,j} \notin \mathcal{M}_u$). Then, we use the following max-margin loss [LeCun *et al.*, 2006] to optimize the ranking order over pairs:

$$L_{m_{u,i} \succeq m_{u,j}} = \max(0, margin + S_{m_{u,j}} - S_{m_{u,i}}) \tag{21}$$

where the parameter $margin$ needs to be tuned over data.

Training Procedure

Our model is implemented using Keras [Chollet, 2015] with Tensorflow as the backend. We initialize the word embeddings with the pre-trained GloVe vectors [Pennington *et al.*, 2014]. Due to the limited space, we only list a brief scheme of the learning procedure on a mini-batch in Algorithm 1, where $\Gamma_{Adam}(\cdot)$ denotes Adam [Kingma and Ba, 2014] based gradient descent optimizer. The code for more detail will be publicly accessible after review.

5 Experiments

The experiments are conducted on the real-world movie watch dataset MovieLens 1M [Harper and Konstan, 2016]. We demonstrate our model from three aspects: (1) recommendation accuracy; (2) new movie recommendation, and (3) interpretation of attraction on movies.

5.1 Data Preparation

We collect user watch records from the MovieLens 1M dataset. However, this dataset does not contain any story and cast data. Fortunately, researchers have provided good mapping from MovieLens ID to DBPedia URI [Noia *et al.*, 2016].

Table 1: Statistics of content-enriched MovieLens dataset

# movies:	3,900	# users:	6,040
# watch record:	1,000,209	# cast:	9,398
movie story vocabulary	22,582	# sentences per story	10.2
# cast members per movie	6.44	# plays per cast	2.10

We queried all available story abstract and cast data from DBPedia. The statistics of the data are reported in Table 1.

For testing the performance on released movie recommendation, we randomly held out 20% user watch records as the testing set, and the remainder were served as the training set. One of the most important tasks is to recommend new movies without knowing any watch record. To simulate this case, we randomly selected 10% movies and held out all their watch records from the dataset, and the remainder of 90% movies and their watch records were used for training. For each hold-out test sample in above two testing sets, we randomly draw ten noisy samples to test whether the testing methods can rank the true sample at a top position out of noisy samples.

5.2 Comparison Methods and Evaluation Metrics

The following state-of-the-art content-based methods are compared for movie recommendation. CF methods cannot deal with rich contents and new movies as the study focus in this paper so they are not included for comparison.

- **CENTROID**: We create user profiles using the centroid [Musto *et al.*, 2016] of all word embedding vectors from the users' movie stories. Then, we rank recommendations by the similarity between the user profile and the controid of word embedding vectors of movie story.
- **CTR**: Collaborative topic regression [Wang and Blei, 2011] performs user regression over the latent topic distribution of movie stories learned from LDA.
- **CWER**: Similar to CTR, we create the collaborative word embedding user regression (CWER) to perform regression over the centroid word embedding vector of each movie story initialized by GloVe embeddings.
- **MLAM**: This is the full multilevel attraction model proposed in this paper.
- **MLAM-S**: This is the single-modal version MLAM that only has the story attraction module.
- **MLAM-C**: This is the single-modal version MLAM that only has the cast attraction module.

To evaluate the recommendation quality, the following metrics are used:

- **R@K**: denotes the mean *Recall* from the top-*K* recommended items over all testing users.
- **MAP@K**: denotes the *Mean Average Precision* of the top-*K* recommended items over all testing users.
- **MRR@K**: denotes the *Mean Reciprocal Rank* of the top-*K* recommended items over all testing users.

Table 2: Ranking performance on released movies (80% training)

Method	MAP@5	MAP@20	MRR@5	MRR@20
CENTROID	0.1738	0.1481	0.0763	0.0958
CTR	0.1226	0.1069	0.0514	0.0692
CWER	0.1666	0.1580	0.0798	0.1089
MLAM-C	**0.4243**	**0.3963**	**0.2118**	**0.2398**
MLAM-S	0.3816	0.3451	0.1822	0.2093
MLAM	**0.4252**	**0.3997**	**0.2187**	**0.2464**

Table 3: Ranking performance on new movies (90% training)

Method	MAP@5	MAP@20	MRR@5	MRR@20
CENTROID	0.2381	0.2409	0.1623	0.1900
CTR	0.1056	0.1374	0.0798	0.1089
CWER	0.1971	0.2346	0.1461	0.1801
MLAM-C	0.1817	0.1664	0.1132	0.1370
MLAM-S	**0.3001**	**0.3059**	**0.2091**	**0.2371**
MLAM	**0.2573**	**0.2671**	**0.1794**	**0.2090**

5.3 Recommendation Accuracy Evaluation

Recommendation for Released Movies

We evaluate the recommendation performance on released movies associated with users' watch records, that is, people have known the story and cast members in these movies. Table 2 reports the recommendation accuracy. CTR scores the user preference according to the topic distribution over a movie story. However, there are many uninformative words which may obscure the core topic distribution of the story. CENTORID and CWER are built on the story embedding derived from centroid of word embeddings. Since word embedding is an unnormalizing vector, it allows large elements to specify the significance. As a result, CENTORID and CWER outperform CTR but they still suffer the obscure from uninformative words. MLAM-S leads CENTORID and CWER with a large margin, the MAP and the MRR are at least 200% higher than baselines. This highlights the design of our model, that is, we place two types of filters in MLAM-S, one is to extract the most attractive words and sentences and the other is to filter out noisy words and sentences. MLAM-C surprisingly performs well, this discloses the fact that the attractiveness of a movie is heavily related to the attractiveness of its cast. Thank to the multimodal modules over story and cast to comprehensively capture users' attraction from different aspects, MLAM achieves the best performance out of all comparison methods.

Figure 3 depicts the recall of all comparison methods. We find that the plots of MLAM-C, MLAM-S and MLAM are above the plots of baselines with apparent margins, i.e., MLAM can more accurately retrieve the attractive movies for each user in top positions. MLAM combines the information from both modules, which leads to the best recall.

Recommendation for New Movies

We apply the above design to recommend attractive new movie, which cannot be handled by pure CF methods, to demonstrate the goal and value of attraction modeling. Content-based methods are more capable of tackling such

User 156	*Attractiveness on sentences*	**Election is a 1999 American comedy-drama film directed and written by Alexander Payne and adapted by him and Jim Taylor from Tom Perrotta's 1998 novel of the same title.** The plot revolves around a high school election and satirizes both suburban high school life and politics. The film stars Matthew Broderick as Jim McAllister, a popular high school social studies teacher in suburban Omaha, Nebraska, and Reese Witherspoon as Tracy Flick, around the time of the school's student body election. When Tracy qualifies to run for class president, McAllister believes she does not deserve the title and tries his best to stop her from winning. Election opened to acclaim from critics, who praised its writing and direction. The film received an Academy Award nomination for Best Adapted Screenplay, a Golden Globe nomination for Witherspoon in the Best Actress category, and the Independent Spirit Award for Best Film in 1999.
	Attractiveness on words	Election is a 1999 American **comedy-drama** film directed and written by Alexander Payne and adapted by him and Jim Taylor from Tom Perrotta's 1998 novel of the same title.
	Attractiveness on cast	**Alexander Payne**, Reese Witherspoon, Matthew Broderick, Jim Taylor
User 2163	*Attractiveness on sentences*	Election is a 1999 American comedy-drama film directed and written by Alexander Payne and adapted by him and Jim Taylor from Tom Perrotta's 1998 novel of the same title. The plot revolves around a high school election and satirizes both suburban high school life and politics. The film stars Matthew Broderick as Jim McAllister, a popular high school social studies teacher in suburban Omaha, Nebraska, and Reese Witherspoon as Tracy Flick, around the time of the school's student body election. When Tracy qualifies to run for class president, McAllister believes she does not deserve the title and tries his best to stop her from winning. Election opened to acclaim from critics, who praised its writing and direction. **The film received an Academy Award nomination for Best Adapted Screenplay, a Golden Globe nomination for Witherspoon in the Best Actress category, and the Independent Spirit Award for Best Film in 1999.**
	Attractiveness on words	The film received an Academy **Award** nomination for **Best** Adapted Screenplay, a Golden Globe nomination for Witherspoon in the **Best** Actress category, and the Independent Spirit **Award** for **Best** Film in 1999
	Attractiveness on cast	Alexander Payne, **Reese Witherspoon**, Matthew Broderick, Jim Taylor

Figure 2: Statistical attractiveness on movie *Election (1999)* w.r.t. sentences, words in the most attractive sentences and cast members from the perspectives of User 156 and User 2163. The larger size and deeper color of font denote the larger attractiveness weight is assigned.

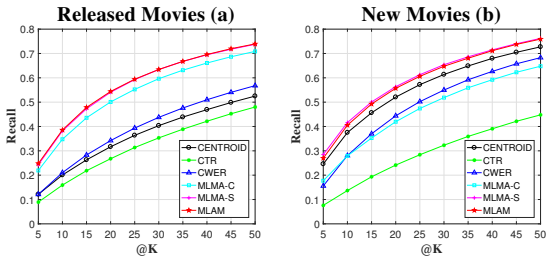

Figure 3: R@5-50 on the Release Movie set and the New Movie set

cases widely seen in this new media age. The ranking performance is reported in Table 3 and the recall is illustrated in Figure 3 (b). CTR underperforms other methods for the reason analyzed in the above subsection. CENTROID and CTR achieve similar performance to the first experiment, which proves the effectiveness of content-based matching using word embedding for new movies even without any user watch record. Similarly, MLAM-S achieves comparable performance with the above case. However, MLAM-C is the special case. We find that the performance drops drastically when comparing with Table 2. In fact, the reason behind is quite straightforward. We can find most cast only appeared in two movies (cf. Table 1). Accordingly, users cannot tell whether they will be attracted by an unknown cast. Figure 4 shows two testing samples of a user. The left movie is associated with a high attractiveness score due to the known cast members (in red) in this user's training set, whereas the cast members in the right movie are absent from user's training set, which results in low attractiveness scores. As a result, MLAM-C tends to assign low rank on these movies. This also proves the factor that the attractiveness of a movie is heavily dependent on its cast members. Accordingly, the multimodal model MLAM slightly underperforms MLAM-S due to the ineffectiveness of MLAM-C.

5.4 Interpretation and Visualization

The most important value of attraction modeling is not only for recommendation but for obtaining insight into the underlying causes of user selection by disclosing the attractive points. In this experiment, we pick two case studies to visualize the user attractiveness scores output by MLAM. Figure 2 illustrates the statistical attractiveness, according to the weights (cf. Eqs. 3, 9 and 15), over the sentences of movie story, words in the most attractive sentence, and the cast members points for User 156 and User 2163. The results show that we can easily find the attraction difference between two users. User 156 is attracted by the first sentence which highlights the genre of this movie, i.e., comedy-drama, while User 2163 is attracted by the last sentence which highlights the award of this movie. Similarly, we find User 156 is attracted by the director Alexander Payne while User 2163 is attracted by the star Reese Witherspoon. Therefore, we can easily use MLAM to analyze user selection and tell the insight about the recommendation made.

Wild America (1997)	***Bogus (1996)***
William Dear, Scott Bairstow, Jonathan Taylor Thomas, Devon Sawa	Norman Jewison, Gérard Depardieu, Whoopi Goldberg, Alvin Sargent Haley Joel Osment

Figure 4: Two comparative testing samples of *User 182*: the left movie *Wild America* obtains a high attraction score because of the cast members in red appear in user's watched movies while the cast members of *Bogus* never appear in user's movie list.

6 Conclusion

In this paper, we propose a multilevel attraction model (MLAM) over multimodal contents to learn user attraction on movies. MLAM can provide the interpretation of user selection w.r.t. the attractive points. Moreover, it can conduct content-based recommendation for new movies. The results prove the effectiveness and merits of MLAM. Textual data, e.g. news, papers and categorical data, e.g. writers, authors, are the most common data in content-based RS, so our model can be directly adapted to these recommendations. For other content like music, image, we can apply CNN to learn their embeddings and then build attraction over them.

Acknowledgement

The paper is partially supported by the National Natural Science Foundation of China (61572325); Shanghai Key Programs of Science and Technology(16DZ1203603); Shanghai Engineering Research Center Project (GCZX14014 and C14001).

References

[Agarwal and Chen, 2010] Deepak Agarwal and Bee-Chung Chen. flda: matrix factorization through latent dirichlet allocation. In *Proceedings of WSDM*, pages 91–100. ACM, 2010.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Balabanović and Shoham, 1997] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[Chollet, 2015] François et al. Chollet. Keras, 2015.

[de Gemmis *et al.*, 2015] Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Semantics-aware content-based recommender systems. In *Recommender Systems Handbook*, pages 119–159. Springer, 2015.

[Denil *et al.*, 2014] Misha Denil, Alban Demiraj, and Nando de Freitas. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*, 2014.

[Harper and Konstan, 2016] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

[Hu *et al.*, 2016] Liang Hu, Longbing Cao, Jian Cao, Zhiping Gu, Guandong Xu, and Dingyu Yang. Learning informative priors from heterogeneous domains to improve recommendation in cold-start user domains. volume 35, page 13. ACM, 2016.

[Hu *et al.*, 2017a] Liang Hu, Longbing Cao, Jian Cao, Zhiping Gu, Guandong Xu, and Jie Wang. Improving the quality of recommendations for users and items in the tail of distribution. *ACM Transactions on Information Systems (TOIS)*, 35(3):25, 2017.

[Hu *et al.*, 2017b] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. Diversifying personalized recommendation with user-session context. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1858–1864. AAAI Press, 2017.

[Kim *et al.*, 2016] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-aware neural language models. AAAI'16, pages 2741–2749. AAAI Press, 2016.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kompan and Bielikova, 2010] Michal Kompan and Maria Bielikova. Content-based news recommendation. In *E-Commerce and Web Technologies: 11th International Conference, EC-Web 2010, Bilbao, Spain, September 1-3, 2010, Proceedings*, volume 61, page 61. Springer Science & Business Media, 2010.

[LeCun *et al.*, 2006] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 1:0, 2006.

[Mcauliffe and Blei, 2008] Jon D Mcauliffe and David M Blei. Supervised topic models. In *NIPS*, pages 121–128, 2008.

[McNee *et al.*, 2006] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, pages 1097–1101, 1125659, 2006. ACM.

[Musto *et al.*, 2016] Cataldo Musto, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. Learning word embeddings from wikipedia for content-based recommender systems. In *European Conference on Information Retrieval*, pages 729–734. Springer, 2016.

[Noia *et al.*, 2016] Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. *ACM Trans. Intell. Syst. Technol.*, 8(1):9:1–9:34, September 2016.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI*, pages 452–461. AUAI Press, 2009.

[Timothy, 2013] Stenovec Timothy. Netflix launches profiles, finally realizing how people really watch movies on it. Aug. 1, 2013.

[Vinyals *et al.*, 2015] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE CVPR*, pages 3156–3164, 2015.

[Wang and Blei, 2011] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD*, pages 448–456, 2020480, 2011. ACM.

[Wang *et al.*, 2017] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *Proceedings of SIGKDD*, 2017.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, 2016.