Modeling User Repeat Consumption Behavior for Online Novel Recommendation

YUNCONG LI*, Tencent, China

CUNXIANG YIN*, Tencent, China

YANCHENG HE, Tencent, China

GUOQIANG XU, Tencent, China

JING CAI, Tencent, China

LEEVEN LUO, Technology zone, China

SHENG-HUA ZHONG[†], Shenzhen University, China

Given a user's historical interaction sequence, online novel recommendation suggests the next novel the user may be interested in. Online novel recommendation is important but underexplored. In this paper, we concentrate on recommending online novels to new users of an online novel reading platform, whose first visits to the platform occurred in the last seven days. We have two observations about online novel recommendation for new users. First, repeat novel consumption of new users is a common phenomenon. Second, interactions between users and novels are informative. To accurately predict whether a user will reconsume a novel, it is crucial to characterize each interaction at a fine-grained level. Based on these two observations, we propose a neural network for online novel recommendation, called NovelNet. NovelNet can recommend the next novel from both the user's consumed novels and new novels simultaneously. Specifically, an interaction encoder is used to obtain accurate interaction representation considering fine-grained attributes of interaction, and a pointer network with a pointwise loss is incorporated into NovelNet to recommend previously-consumed novels. Moreover, an online novel recommendation dataset is built from a well-known online novel reading platform and is released for public use as a benchmark. Experimental results on the dataset demonstrate the effectiveness of NovelNet [1].

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: online novel recommendation, repeat consumption, interaction understanding

1 INTRODUCTION

Online novel reading platforms (e.g. QQ browser, SOFANOVEL and Wattpad) provide online novels for reading and are increasingly becoming a major part of people's daily entertainments. As of November 2021, Wattpad has 90

---

*Both authors contributed equally to this research.
[†]Corresponding author.
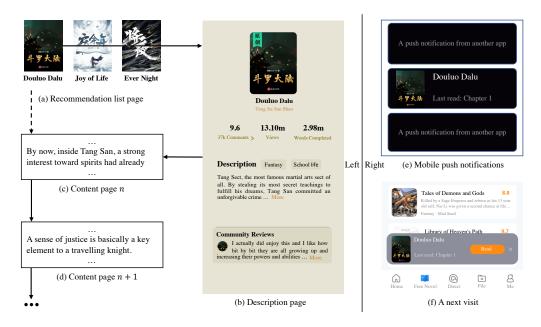[1]Data and code are released at https://github.com/l294265421/NovelNet

Fig. 1. Each subfigure is part of a phone screen shot. (1) The left part is an example illustrating an interaction between a user and a novel. The solid line arrows indicate that the user goes from one page to another. The dotted line arrow indicates that the user can go from the recommendation list page to the content page directly. The interaction begins when the user goes from the recommendation list page to a page of the novel. The interaction terminates when the user leaves the novel and returns to the recommendation list page, and the user can leave from any page of the novel. (2) The right part shows two important online novel recommendation scenarios where recommending users' previously-consumed novels plays a key role.

million monthly users [2]. The growing users and the increasing volume of online novels suggest the need of effective online novel recommender systems for such platforms. Although online novel recommendation is important, it is still underexplored. To the best of our knowledge, only a tag-driven algorithm with collaborative item modeling was proposed for online novel recommendation [21]. Moreover, although conventional book recommendation has been studied by many researchers [12, 27, 34, 38], it differs from online novel recommendation. One key difference between conventional book recommendation and online novel recommendation lies in whether or not recommendation models need to model the process of users reading each individual novel. The difference is from the platforms these models serve. While the platforms (e.g. BookCrossing, Amazon and GoodReads) that conventional book recommendation serves only help users to discover books, online novel reading platforms additionally allow users to read novels on their platforms. This difference leads to that online novel recommendation models need to model the process of users reading novels, while conventional book recommendation models do not.

In this paper, we concentrate on recommending novels to new users. On a particular day, the new users are defined as users whose first visits to an online novel reading platform happened in the last seven days. As the new users are not familiar with the online reading platform, it may be difficult for them to find the novels they are interested in. Therefore, an effective online novel recommender system is especially essential for these new users. Moreover, online novel reading platforms have spent millions of dollars on advertising to attract these new users to their platforms, thus

---

[2]https://en.wikipedia.org/wiki/Wattpad

effective online novel recommendation models are urgent for these platforms to make these new users stay on their platforms.

We have two observations about online novel recommendation for new users. First, repeat novel consumption of new users is a common phenomenon. Specifically, the repeat ratios of the training, validation and test sets of the online novel recommendation dataset discussed in section 4.1 are 58.31%, 59.48% and 56.44%, respectively. Therefore, it is necessary to recommend previously-consumed novels to users. In fact, new users usually do not read a whole novel multiple times. Instead, users usually require much time to finish reading a novel, hence cannot read a novel all at one sitting and will interact with a novel many times to finish reading it. **That is, in this paper, consuming a novel means interacting with a novel one time rather than reading a novel all.** Second, interactions between users and novels are informative. An example of interaction is shown in the left part of Fig. 1. It is intuitive that different kinds of interactions between a user and a novel can reflect the user's different preferences for the novel. For example, a user reads the description page, then returns to the recommendation list page, which may indicate that the user is not interested in the novel and hence will not interact with the novel again. A user reads the description page, then reads many content pages, which may show the user prefers this novel and will interact with it in the future. Therefore, to accurately predict whether a user will reconsume a novel, it is crucial to characterize each interaction at a fine-grained level.

To make our motivation of recommending novels that new users have already known clearer, we additionally describe two important online novel recommendation scenarios. The first scenario is that an online novel reading platform will send a mobile push notification to a new user when the new user does not actively visit the platform [3]. The notification presents a novel that the new user may want to read. By clicking the notification, the new user can directly enter the platform and start reading the novel. The novel can be a new novel or a novel the user has already known. Note that it is common that new users do not come back to the platform after their first visits. An example of mobile push notifications is shown in Fig. 1 (e). The second scenario is that an online novel reading platform may present a previously consumed novel to a new user when the new user next enter the platform. The purpose is to provide a shortcut to the new user since new users may be not familiar with the online novel reading platform and hence cannot find the preferred novels again easily. What's more, in this scenario, online novel reading platforms will not present the recommended novel if the recommender system suggests a new novel. An example of this scenario is shown in Fig. 1 (f).

Based on the two observations mentioned above, we propose a neural network for online novel recommendation, called NovelNet. Given a user's historical interaction sequence, NovelNet recommends the next novel the user may prefer from both the user's consumed novels and new novels simultaneously. Specifically, fine-grained interaction attributes are extracted to characterize interaction. An interaction encoder is used to obtain accurate interaction representation considering these fine-grained attributes of interaction. The contextualized representation of the last interaction with a novel summarizes all interactions with the novel and is used to model the whole process of the user reading the novel. Since recommending a novel from a user's consumed novels can be treated as selecting a position from the user's interaction sequence, which is just the problem pointer networks [33] attempt to solve. Thus, a pointer network is incorporated into NovelNet to recommend consumed novels. Since original pointer networks [33] can only learn the relative order of consumed novels, a pointwise loss [24] is added and attempts to learn real interaction probability.

The contributions of this work can be summarized as follows:

- We are the first to explore user repeat consumption behavior in online novel recommendation.

---

[3]Mobile push notification is a very effective channel for online services to engage with users and drive user engagement metrics [37].

- We propose NovelNet for online novel recommendation, which encodes interaction considering fine-grained interaction attributes and uses a pointer network with a pointwise loss to model the user repeat consumption behavior.
- An online novel recommendation dataset is built from a well-known online novel reading platform and is released for public use as a benchmark. Experiments on the dataset show the effectiveness of our method.

2 RELATED WORK

Online novel recommendation is underexplored. As far as we know, only Li et al. [21] proposed a tag-driven algorithm with collaborative item modeling (TDCIM) for online novel recommendation. Li et al. [21] observed that the majority of users only consume a few type of novels over a certain period. However, there are broad categories of novels in the initial recommendation list achieved by previous collaborative filtering models. To solve this issue, TDCIM exploits novel tags to lower the rankings of uninteresting categories and raise those of interesting categories. However, Li et al. [21] didn't consider repeat novel consumption behavior of users and the dataset is not publicly available. Furthermore, different from [21], this work focuses on new users.

Repeat consumption has been studied in various domains, such as, E-commerce [5, 20, 30, 35], music listening [19, 29, 30], live-streaming [28], web revisitation [1, 23], and repeated web search queries [31, 32]. However, repeat consumption has not been explored yet in online novel reading. Additionally, the main reason of repeat consumption in online novel reading differs from that in the above-mentioned domains. In online novel reading, users interact with a novel multiple times mainly because finishing reading a novel requires much time. However, in the domains mentioned above, take E-commerce as an example: people purchase a product multiple times mainly because the product is consumable. From a method perspective, previous methods can be separated into three broad categories: i) models that predict whether an interaction will be a repeat consumption [7], ii) those that predict which consumed item users will prefer given the fact that current consumption is a repeat consumption [2, 5, 8], and iii) models that simultaneously recommend new items and previously-consumed items [30]. Our work belongs to the third category. Compared with previous models in the third category, our method takes more interaction attributes as input considering the special characteristics of online novel recommendation of new users.

Book recommendation has been explored by many works. In order to reflect the user's complete spectrum of interests in books, Ziegler et al. [38] proposed topic diversification to balance and diversify personalized recommendation lists. McAuley et al. [27] predicted which books will be co-purchased based on their cover art. Wan and McAuley [34] arranged a user's different behavior (e.g. rate, review) towards a book in a chain. Given historical observations of users' behavior chains, Wan and McAuley [34] sought to estimate their behavior chains toward unobserved items. Ekstrand and Kluver [12] examined the response of collaborative filtering recommender algorithms to the distribution of their input data with respect to book creator gender. The book datasets used in these works include BookCrossing [38], Amazon Books [27] and GoodReads [34]. BookCrossing was collected from the reading community BookCrossing [4], Amazon Books was gathered from the E-commerce platform Amazon [5], and GoodReads was collected from the book discovery service GoodReads [6]. These platforms, BookCrossing, Amazon and GoodReads differ from online novel reading platforms. While BookCrossing, Amazon and GoodReads only help users to discover new books, online novel reading platforms additionally allow users to read novels on their platforms. The difference leads to that book recommendation

---

[4]https://www.bookcrossing.com/
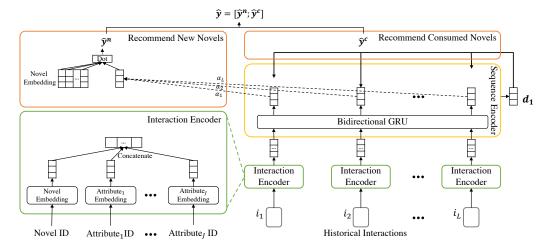[5]https://www.amazon.com/
[6]https://www.goodreads.com/

Fig. 2. The architecture of our NovelNet for online novel recommendation.

in existing literature differs from online novel recommendation.Specifically, online novel recommendation needs to model the process of a user reading a novel and then predicts whether the user will resume reading the novel the user has already known, while previous book recommendation does not.

Session-based recommendation [36] aims to suggest items for anonymous users usually with a short interaction sequence. Anonymous users can be seen as new users. Moreover, session-based recommendation models also take the user's historical interactions as input and predict an item the user may be interested in next. Therefore, the setting of session-based recommendation is similar to ours. The core difference is that, the interaction in session-based recommendation is usually just item id, while the interaction in online novel recommendation contains item id and other interaction attributes. Most importantly, a few session-based recommendation models [11, 30] also consider user repeat consumption behavior. Hence, important or state-of-the-art session-based recommendation models are selected as baselines in this work.

3 METHOD

In this section, we first introduce the task definition, then describe our NovelNet for online novel recommendation.

3.1 Task Definition

Let U be the set of all users and N be the set of all novels. Each user $u \in U$ has an interaction sequence $S_u = \{i_1, i_2, ..., i_L\}$. The sequence is sorted by time in an ascending order and each interaction has several interaction attributes discussed later. A novel may appear in S_u more than one time. All novels appearing in S_u are denoted by N^c. $N^n = N - N^c$ includes all novels that the user has not interacted with. Given a target time t and a user u with S_u, NovelNet predicts the novel $N_t \in N$ the user is most likely to interact with at the target time.

3.2 NovelNet

Our NovelNet contains four major modules: Interaction Encoder, Sequence Encoder, Recommend New Novels and Recommend Consumed Novels. The architecture of NovelNet is shown in Fig. 2.

3.2.1 Interaction Encoder. Intuitively, to model the process of a user reading a novel and then accurately predict whether the user will resume reading the novel, it is critical to represent the interactions between the user and the novel at a fine-grained level. In order to characterize interactions between users and novels, several interaction attributes are mined. It is intuitive that if a user i) reads the description page and then continues to read content pages, or ii) reads many content pages, or iii) adds the novel to his/her library, or iv) spends much time reading the novel, or v) has interacted with the novel multiple times, the user may be interested in the novel and willing to resume reading the novel. Based on these intuitions, five interaction attributes are extracted as follows:

- description_content (a^1): if the user first reads the description page, then reads at least one content page, the value of this attribute is 2, otherwise 1.
- real_read (a^2): if the user reads at least two content pages, the value of this attribute is 2, otherwise 1.
- collect (a^3): if the user adds the novel to his/her own library, the value of this attribute is 2, otherwise 1.
- read_duration (a^4): minutes that this interaction lasts.
- novel_count (a^5): the number of times the novel appears in the user's history until this interaction.

Besides the interaction attributes above, we also explore four attributes which prove to be useful for predicting reconsumption in other domains. Specifically, Anderson et al. [2] showed that *recency* and *quality* are beneficial to reconsumption prediction, since recently consumed items are likely to be consumed again and high-quality items are more likely to be consumed. Benson et al. [4] observed that increasing boredom of an item leads up to eventual abandonment of the item and used the gaps between a user's consumption of the same item to represent the boredom of the item. Benson et al. [4] explored two types of gap, i.e. *temporal_gap* and *index_gap*. In online novel recommendation, the four attributes are defined as follows:

- recency [2] (a^6): the difference in hours between the target time and the start time of this interaction.
- quality (a^7): following Chen et al. [7], we use popularity to measure the quality of each novel. The novel popularity is defined as the natural logarithm (base e) of the frequency of the novel in the training set.
- temporal_gap (a^8): the difference in hours between the start time of the last interaction with this novel and the start time of this interaction. If the user first interacts with the novel, the value is set to 0.
- index_gap (a^9): the number of novels which appear between the last interaction with this novel and this interaction. If the user first interacts with the novel, the value is set to 0.

Moreover, novel ID is used to track novels. Since novel tags are useful for discovering new novels for users [21], hence is also treated as an interaction attribute and is explored.

- novel (a^{10}): the novel ID.
- novel_tag (a^{11}): the tag ID the novel is associated with. An example of novel tags is Romance.

For each attribute, if its value is not an integer, the value will be converted into an integer which is the smallest integer greater than or equal to the value. For attribute a^j, the interaction encoder converts its value into a vector e^j via an embedding look-up table $W_e^j \in R^{V^j \times D^j}$, where V^j and D^j are vocabulary size and embedding dimension respectively. These attributes use different embedding look-up tables. Then, we obtain a set of vectors $[e^1, e^1, ..., e^{11}]$. The interaction representation r is the concatenation of these attribute vectors, $r = [e^1; e^2; ...; e^{11}]$. Finally, the interaction sequence S_u is converted into a sequence of vectors $[r_1, r_2, ..., r_L]$ by the interaction encoder.

3.2.2 Sequence Encoder. A recent empirical analysis [26] in session-based recommendation showed that GRU [10] is highly competitive among neural models. In addition, in our pilot experiments, bidirectional GRU achieves better

performance than unidirectional GRU. Thus the sequence encoder we use is bidirectional GRU. The bidirectional GRU takes the sequence of interaction vectors $[r_1, r_2, ..., r_L]$ as input, then outputs forward vectors $\overrightarrow{H} = [\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_L}]$ and backward vectors $\overleftarrow{H} = [\overleftarrow{h_1}, \overleftarrow{h_2}, ..., \overleftarrow{h_L}]$. The concatenation of $\overrightarrow{H}$ and $\overleftarrow{H}$ is $H = [h_1, h_2, ..., h_L]$, where $h_l = [\overrightarrow{h_l}; \overleftarrow{h_l}]$.

3.2.3 Recommend New Novels.
Given the output of the sequence encoder, $\overrightarrow{H}$, $\overleftarrow{H}$ and H, we first use an attention mechanism [3] to generate the sequence representation:

$$u_l^n = (v^n)^T tanh(W_1^n h_l + W_2^n [\overrightarrow{h_L}; \overleftarrow{h_1}]), l \in (1, 2, ..., L) \tag{1}$$

$$\alpha = softmax(u^n) \tag{2}$$

$$r^{S_u} = \sum_{l=1}^{L} \alpha_l h_l \tag{3}$$

where v^n, W_1^n and W_2^n are learnable parameters, and u_l^n (α_l) is the l-th entry of u^n (α).

Given a novel k from N^n, the interaction score is calculated by the inner product of the representation vector of the sequence r^{S_u} and the novel embedding, i.e., $s_k = (r^{S_u})^T (W_e^{10})_k$. The normalized score is $\hat{y}^n = softmax(s)$, where s_k is the k-th entry of s. The loss function for recommending new novels is the negative log-likelihood of the ground truth novel:

$$\mathcal{L}^n = - \sum_{k \in N^n} y_k^n log(\hat{y}_k^n) \tag{4}$$

where if k is the ground truth novel, $y_k^n = 1$, otherwise $y_k^n = 0$. And $\hat{y}_k^n$ is the l-th entry of $\hat{y}^n$.

3.2.4 Recommend Consumed Novels.
A pointer network [33] with only one decoding step is used to recommend consumed novels. Given the output of the sequence encoder, $\overrightarrow{H}$, $\overleftarrow{H}$ and H, the first decoder hidden state is obtained by $d_1 = [\overrightarrow{h_L}; \overleftarrow{h_1}]$. Then the pointers are computed as:

$$u_l^c = (v^c)^T tanh(W_1^c h_l + W_2^c d_1), l \in (1, 2, ..., L) \tag{5}$$

$$p^{listwise} = softmax(u^c) \tag{6}$$

$$\hat{y}_l^c = p_l^{pointwise} = sigmoid(u_l^c) \tag{7}$$

where v^c, W_1^c and W_2^c are learnable parameters and u_l^c ($p_l^{pointwise}$ or $\hat{y}_l^c$) is the l-th entry of u^c ($p^{pointwise}$ or $\hat{y}^c$).

The $p^{listwise}$ and $p^{pointwise}$ are used to compute listwise loss and pointwise loss [24], respectively:

$$\mathcal{L}_{listwise}^c = - \sum_{l=1}^{L} m_l y_l^c log(p_l^{listwise}) \tag{8}$$

$$\mathcal{L}_{pointwise}^c = -\frac{1}{L} \sum_{l=1}^{L} m_l(y_l^c log(\hat{y}_l^c) + (1 - y_l^c)log(1 - \hat{y}_l^c)) \tag{9}$$

where if the novel associated with the l-th interaction is the ground truth novel, $y_l^c = 1$, otherwise $y_l^c = 0$. For the novel associated with the l-th interaction, if the l-th interaction is the last interaction with the novel, $m_l = 1$, otherwise $m_l = 0$. The motivation behind this is that the contextualized representation of the last interaction with a novel summarizes all interactions with the novel and is used to model the whole process of the user reading the novel. We call m_l the mask of the l-th interaction. While in the original pointer networks [33] only $p^{listwise}$ is computed and used to produce output, $p^{pointwise}$ is additionally computed and used to select the next novel by us. The reason is that the values in $p^{listwise}$ only indicate the relative order of the consumed novels rather than the real interaction probability and hence cannot be

Table 1. Detailed statistics of our dataset. #Description_content (#Real_read, #Collect) represents the number of interactions whose description_content (real_read, collect) attribute values are 2. A and M are the average and median interaction numbers of users, respectively. RR stands for repeat ratio (%), which is defined as the number of instances corresponding to previously-consumed novels for all users divided by the total number of instances for all users. An instance is an interaction which is not the first interaction of the user.

Dataset	#user	#novel	#interaction	#instance	#description _content	#real _read	#collect	A	M	RR
training	91311	18487	602067	548880	133866	333010	119996	9.45	4	58.31
valid	34857	10598	121384	111473	21904	65948	26870	9.71	5	59.48
test	38338	10265	121433	111859	20816	62895	31255	9.18	4	56.44

directly compared with the values in $\hat{\boldsymbol{y}}^n$ (i.e. the predictions of new novels). For example, if the user interacted with only one novel, i.e. $L = 1$, no matter whether or not the user is interested in the novel, $\boldsymbol{p}^{listwise}$ only has one entry with value 1 which is the predicted probability the user will interact with the novel. This is unreasonable. To mitigate this issue, we use $\boldsymbol{p}^{pointwise}$ to learn real interaction probability. The listwise loss $\mathcal{L}^c_{listwise}$ is kept, because it can improve model performance.

3.2.5 Training and Inference. We jointly train NovelNet. The combined loss function is:

$$\mathcal{L} = \lambda \mathcal{L}^c_{pointwise} + \mathcal{L}_{listwise} \tag{10}$$

where λ is the weight of pointwise loss. If the ground truth novel is from N^n, $\mathcal{L}_{listwise} = \mathcal{L}^n$, otherwise $\mathcal{L}_{listwise} = \mathcal{L}^c_{listwise}$. During inference, the l-th entry $\hat{y}^c_l$ of $\hat{\boldsymbol{y}}^c$ is set to 0, if the corresponding interaction mask $m_l = 0$. We then obtain the interaction probability scores $\hat{\boldsymbol{y}}$ of all novels in N by concatenating $\hat{\boldsymbol{y}}^n$ and $\hat{\boldsymbol{y}}^c$, i.e., $\hat{\boldsymbol{y}} = [\hat{\boldsymbol{y}}^n; \hat{\boldsymbol{y}}^c]$, finally select the novel with the highest score.

4 EXPERIMENTS

4.1 Dataset and Metrics

Since Li et al. [21], the only work we found on online novel recommendation, did not release their dataset, instead, we constructed an online novel recommendation dataset based on the logs collected from QQ browser[7], a famous online novel reading platform belonging to Tencent[8] in China, during 14 days (from Nov. 11, 2021 to Nov. 24, 2021). Only the logs of new users appearing between Nov. 18 2021 and Nov. 24, 2021 were kept and a subset of all the new users was randomly sampled to build our dataset. The interactions between Nov. 18 2021 and Nov. 22, 2021 were used for training. The interactions on Nov. 23, 2021 and Nov. 24, 2021 were used for validation and test respectively. The interactions before Nov. 18 2021 were kept and used for models' input, which guaranteed that models can use all historical interactions of a user to predict the user's next interaction. We filtered out users with only 1 interaction. Novels that appeared less than 5 times in the training set were also removed. The detailed statistics of this dataset are summarized in Table 1 and Fig. 3.

Following previous work [30], we use MRR and Recall as metrics. Besides MRR@10, MRR@20, Recall@10 and Recall@20, we also report MRR@1[9], MRR@5 and Recall@5. MRR@1 is the main evaluation metric, since in some

---

[7]https://browser.qq.com/
[8]https://www.tencent.com/
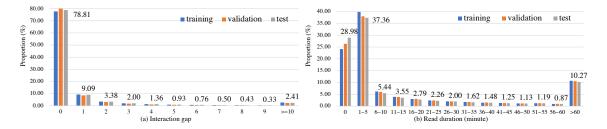[9]Recall@1 is equal to MRR@1, therefore, only MRR@1 is reported

Fig. 3. The histograms of interaction gap and read duration in our dataset. For clarity, only the statistics of the test set are marked on the corresponding bars. Interaction gap is defined as the number of interactions between consumption of the same novel. For example, interaction gap 0 indicates that this interaction and the last interaction of a user relate to the same novel. In (a), the proportion is the number of the reconsumption interactions with the corresponding interaction gap to all reconsumption interactions. An interaction is a reconsumption interaction if the user has interacted with the novel of the interaction before. In (b), the proportion is the number of the interactions whose read_duration attribute values are within the corresponding interval to all interactions.

important online novel recommendation scenarios, as shown in the right part of Fig. 1, only one recommended novel has a chance to be shown to the user.

- MRR@k (Mean Reciprocal Rank) is the average of reciprocal ranks of the desired novels. The reciprocal rank is set to zero if the rank is larger than k.
- Recall@k is the proportion of cases when the desired novel is among the top-k novels in all test cases.

4.2   Implementation Details

In our experiments, the embedding dimension is set to 128 for novel and 32 for the other interaction attributes. The GRU hidden state size is set to 64. We use Adam [18] as the optimizer (lr=0.001). The pointwise loss weight λ is set to 12. In our final model, only a subset of the interaction attributes introduced in section 3.2.1 is included, including novel, description_content, real_read, read_duration, novel_count, recency, and temporal_gap, since the other attributes cannot improve model performance or are inferior to their counterparts. These hyperparameters are tuned on the validation set. We run all models for 5 times. Each time, the model with highest MRR@1 on the validation set is evaluated on the test set . The average results of all models on the test set are reported.

4.3   Performance Comparison

We compare the proposed NovelNet with several representative methods in session-based recommendation. (1) six non-neural-network-based models: AR [25], and SR [25], SKNN [17], VSKNN [25], STAN [13], and VSTAN [26]; (2) NARM [22] and GRU4REC+ [14, 15][10]; (3) RepeatNet [30], SLIST [11] and Proxy-SR [9]. The reason is that a recent empirical analysis [26] in session-based recommendation showed that non-neural-network-based models provided more accurate recommendations than other neural architectures and NARM as well as GRU4REC+ were highly competitive among neural models. SLIST and Proxy-SR are two state-of-the-art models in session-based recommendation. RepeatNet and SLIST consider repeat consumption. In these methods, each interaction only includes novel ID. All these methods recommend the next novel from all novels (i.e. N). Moreover, we also compare NovelNet with a rule-based method: RecentNovel, which always recommends the novel users last interacted with.

---

[10]GRU4REC+ uses BPR-max loss and is an improved version of original GRU4REC [16]

Table 2. Performance comparison of different methods. The best scores are in bold and the second best scores are underlined. The best scores are significantly better than the corresponding second best scores in paired t-test ($p < 0.05$).

Method	MRR (%)				Recall (%)		
	@1	@5	@10	@20	@5	@10	@20
AR	40.36	43.09	43.52	43.89	47.86	51.18	56.55
SR	42.58	45.13	45.70	46.06	49.85	54.10	59.50
SKNN	32.84	39.71	40.65	41.00	51.19	58.16	63.13
VSKNN	41.97	46.85	47.49	47.80	54.52	59.37	63.81
STAN	43.06	48.78	49.50	49.84	<u>57.81</u>	<u>63.11</u>	<u>68.11</u>
VSTAN	43.88	48.62	49.25	49.51	55.99	60.63	64.32
NARM	42.90	47.27	48.01	48.43	54.72	60.25	66.34
GRU4REC$^+$	42.20	45.64	46.28	46.68	51.81	56.62	62.20
RepeatNet	<u>44.93</u>	<u>49.52</u>	<u>49.99</u>	<u>50.36</u>	56.42	60.03	65.56
SLIST	38.66	44.07	44.81	45.18	53.05	58.51	63.88
Proxy-SR	41.69	46.01	46.79	47.14	53.44	59.30	64.36
RecentNovel	44.48	-	-	-	-	-	-
NovelNet	**47.02**	**51.33**	**52.00**	**52.37**	**58.36**	**63.43**	**68.72**

The experimental results are shown in Table 2. First, we observe that, consistent with [26], STAN surpasses other non-neural-network-based models (one exception is VSTAN on MRR@1) and NARM as well as GRU4REC$^+$. Second, RecentNovel obtains better MRR@1 than all baselines except RepeatNet. The reason is that resuming reading a novel is a common phenomenon and a lot of reconsumption behavior relates to the novel users last read (Table 1 and Fig. 3 (a)). Third, RepeatNet outperforms all other baselines in terms of MRR@1, indicating the necessity to explicitly model the repeat novel consumption. Although SLIST also models repeat consumption, it is a linear model and thus has insufficient model capacity for online novel recommendation. Fourth, the proposed NovelNet outperforms all baselines, indicating the effectiveness of the combination of the pointer network with a pointwise loss and our mined interaction attributes. In addition, while RepeatNet obtains better MRR scores than STAN, STAN obtains better Recall scores. Compared with STAN, NovelNet achieves bigger gains in terms of MRR than in terms of Recall. The possible reason is that the recommendation lists of STAN also include users' previously-consumed novels, but the desired novels have low ranks.

4.4 Impact of Interaction Attribute

In this section, we explore the effects of the interaction attributes. For all models, the pointwise loss weight λ is set to 1. Experimental results are shown in Fig. 4, where NovelNet-160 as well as NovelNet-320 represent NovelNet only using novel ID (160 and 320 indicate novel embedding dimensions) and NovelNet-${attribute_name} using novel ID and only one additional attribute, i.e. ${attribute_name}. Through increasing the novel embedding dimension, the interaction representation size in NovelNet-160 is the same as that in models using only one additional interaction attribute, and the interaction representation size in NovelNet-320 is the same as that in NovelNet. First, we observe that NovelNet-160 and NovelNet-320 have similar performance, which shows simply increasing the novel embedding dimension cannot improve model performance. Second, NovelNet-description_content, NovelNet-real_read and NovelNet-read_duration obtain better performance than the other models using only one additional attribute and NovelNet-160, indicating that the interaction attributes mined by considering the special characteristics of online novel reading are more powerful. Third, NovelNet-novel_count, NovelNet-recency and NovelNet-temporal_gap also significantly outperform NovelNet-160, therefore are included by our final NovelNet. Fourth, the attribute collect cannot improve model performance. One
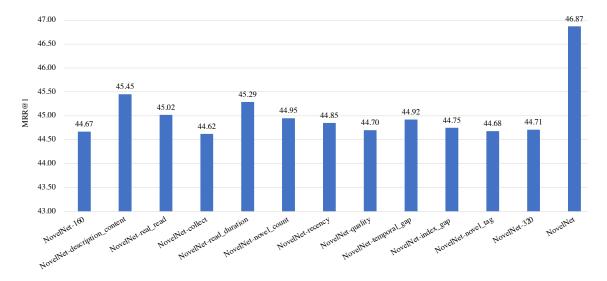
Fig. 4. Impact of interaction attributes in terms of MRR@1 (%).

possible reason is that some collection action (i.e. adding novels to users' own libraries) is taken by the platform as opposed to by the users themselves [11] and hence the attribute collect is noisy. Fifth, NovelNet-quality as well as NovelNet-novel_tag do not significantly surpasses NovelNet-160 and NovelNet-index_gap is inferior to NovelNet-temporal_gap, thus the attributes quality, novel_tag and index_gap are not included by our final NovelNet. In addition, NovelNet significantly outperforms the models using only one additional attribute, indicating that combining more effective interaction attributes can further improve model performance and hence it is essential to accurately characterize interactions.

To further observe where the performance improvement is from, we divided the MRR@1 scores of the models into two parts: consumed and new. While the consumed part is computed only considering the correct predictions on previously-consumed novels, the new part is computed only considering the correct predictions on new novels. For a model, its MRR@1 is equal to the consumed part plus the new part. For example, the MRR@1 of NovelNet is 46.87% and the corresponding consumed and new parts are 46.65% and 0.42% respectively. The division results are shown in Table 3. We draw the following conclusions from Table 3. First, for all models, the consumed part is much bigger than the new part, indicating that the MRR@1 scores are mainly from the correct predictions on previously-consumed novels. Second, interaction attributes can improve both the consumed part and the new part, but the most gain is from the consumed part.

4.5 Impact of Pointwise Loss

4.5.1 Impact of Pointwise Loss Weight. We first explore the impact of the pointwise loss weight λ. We vary λ ranging from 1 to 15. The MRR@1 scores on the validation set and the test set are shown in Fig. 5. From the results on the validation set, we can see that the scores grow with the weight when the weight is less than or equal to 12. The scores stop increasing when the weight is bigger than 12. Therefore, 12 is the weight used by our final NovelNet.

---

[11]On the online novel reading platform that we collect our dataset from, when some predefined conditions are satisfied, the novel will added to the user's library by the platform.

Table 3. The division results of the MRR@1 scores (%). The results of the models using the attributes which are not included by our final NovelNet are not shown there.

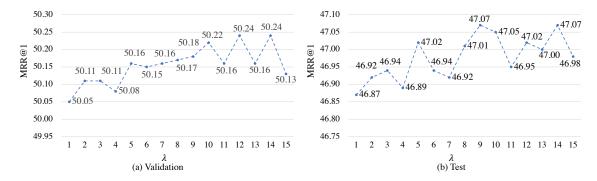| Type	NovelNet -160	NovelNet -description _content	NovelNet -real _read	NovelNet -read _duration	NovelNet -novel _count	NovelNet -recency	NovelNet -temporal _gap	NovelNet
Consumed	44.67	45.42	44.93	45.16	44.89	44.76	44.88	46.45
New	0	0.03	0.09	0.12	0.06	0.09	0.04	0.42

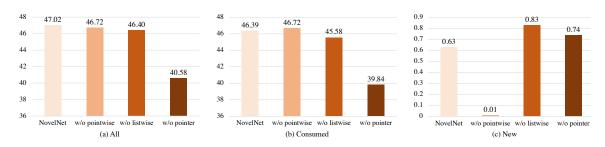Fig. 5. Impact of pointwise loss weight in terms of MRR@1 (%).

Fig. 6. Impact of pointwise loss in terms of MRR@1 (%).

4.5.2 Impact of Pointwise Loss. Then we explore the effect of the pointwise loss. Specifically, we explore three variants of NovelNet: (1) w/o pointwise which removes the pointwise loss $\mathcal{L}^c_{pointwise}$ and sets $\hat{y}^c = p^{listwise}$ (2) w/o listwise which removes the listwise loss $\mathcal{L}^c_{listwise}$, and (3) w/o pointer which removes the pointer network and uses Recommend New Novels module to predict the next novel from all novels[12]. Experimental results are shown in Fig. 6 (a). We observe that NovelNet obtains better performance than both w/o pointwise and w/o listwise, indicating that both pointwise loss and listwise loss are helpful. Moreover, w/o pointer is greatly inferior to NovelNet, indicating the necessity to explicitly model repeat novel consumption behavior of new users and the effectiveness of the pointer network with a pointwise loss.

Similar to section 4.4, the MRR@1 scores of NovelNet, w/o pointwise, w/o listwise and w/o pointer are separated into two parts: consumed and new, and the results are shown in Fig. 6 (b) and (c). We can see that the reason that NovelNet

---

[12]w/o pointer differs from GRU4REC+ in Table 2. While w/o pointer uses cross-entropy loss, GRU4REC+ uses BPR-max loss.

Table 4. The impact of the interaction mask.

Method	MRR (%)				Recall (%)		
	@1	@5	@10	@20	@5	@10	@20
NovelNet	**47.02**	**51.33**	**52.00**	**52.37**	**58.36**	**63.43**	**68.72**
NovelNet - w/o mask	45.84	50.38	50.99	51.35	57.53	62.11	67.40

Table 5. The impact of the repeat ratios of recommendation results on the new part of MRR@1 scores.

Metric	NovelNet	NovelNet ($\lambda = 1$)	w/o pointwise	w/o listwise	w/o pointer	RepeatNet
Repeat ratio of recommendation results (%)	90.27	93.12	99.95	86.78	71.11	97.39
New part of MRR@1 (%)	0.63	0.42	0.01	0.83	0.74	0.07

Table 6. Performance of the Recommend New Novels and Recommend Consumed Novels modules in terms of MRR@1 (%).

Module	NovelNet	NovelNet ($\lambda = 1$)	w/o pointwise	w/o listwise	w/o pointer
Recommend New Novels	3.67	3.37	2.94	3.98	3.97
Recommend Consumed Novels	82.94	82.91	82.79	82.21	76.60

surpasses w/o pointwise is because NovelNet makes more correct predictions on the new part with the help of the pointwise loss.

4.6 Impact of Interaction Mask

In this section, we explore the impact of the interaction mask, i.e. m_l in Equation 8 and 9. By removing m_l in Equation 8 and 9, we obtain a variant of NovelNet, NovelNet - w/o mask . Table 4 shows the performance of NovelNet and NovelNet - w/o mask . We can see that NovelNet surpasses NovelNet - w/o mask across all metrics. A possible reason is that NovelNet only using the contextualized representation of the last interaction with a novel can model the whole process of the user reading the novel more easily.

4.7 Why Is the New Part of MRR@1 of models small

From Table 3 and Fig. 6, we observe that the new part of MRR@1 of all models is very small. To find possible reasons, we first analyze the repeat ratios of the recommendation results of models. The repeat ratio of the recommendation results of a model is defined as the number of predicted novels with highest scores coming from previously-consumed novels divided by the total number of interactions needed to be predicted. The repeat ratios of several models are shown in Table 5. Table 5 also includes the new part of MRR@1 scores of these models. From Table 5, we draw following three conclusions. First, the repeat ratios of the recommendation results of these models are very big and much higher than the repeat ratio of the test set. That is, these models has the popularity bias problem [6]. Consequently, new novels have slight chances to be recommended, resulting in small new part of MRR@1 scores. Second, among the methods explicitly modelings repeat novel consumption (i.e. all models except w/o pointer), NovelNet and w/o listwise have smaller repeat ratios and hence their new part of MRR@1 is bigger with the help of pointwise loss and bigger pointwise

loss weight. Third, although w/o pointer has smaller repeat ratio than w/o listwise, w/o listwise obtains bigger new part of MRR@1 than w/o pointer, which further shows the effectiveness of pointwise loss.

Another possible reason that the new part of MRR@1 of models is small may be that predicting new novels is harder than predicting consumed novels. Therefore, we investigate the performance of the Recommend New Novels and Recommend Consumed Novels modules. Specifically, when evaluating the performance of the Recommend New Novels module, we only consider the instances whose ground truth is a new novel and only recommend new novels. When evaluating the performance of the Recommend Consumed Novels module, we only consider the instances whose ground truth is a consumed novel and only recommend consumed novels. The results are shown in Table 6. We can see that the MRR@1 scores of the Recommend New Novels module of all models are smaller than 4%, while the MRR@1 scores of the Recommend Consumed Novels module of all models are bigger than 75%. This indicates that predicting new novels is indeed more difficult than predicting consumed novels. One reason is that the number of candidates of the Recommend Consumed Novels module is much larger than that of the Recommend New Novels module.

5   CONCLUSION

In this paper, we tackle the next-item recommendation in online novel domain. Specifically, we observe that repeat novel consumption of new users is common and accurately characterizing the interaction is important for modeling repeat novel consumption. Thus, we propose NovelNet, which encodes interaction considering fine-grained interaction attributes and uses a pointer network with a pointwise loss to model the user repeat consumption behavior. An online novel recommendation dataset is built from an online novel reading platform and is released for public use as a benchmark. Experiments on the dataset show the effectiveness of our method. However, as shown in the Experiments section, our model has the popularity bias problem. In the future, we will try to mitigate the popularity bias problem of NovelNet. Moreover, NovelNet only uses the contextualized representation of the last interaction with a novel to implicitly model the whole process of the user reading the novel, which may be not effective enough. We will explicitly model the reading process in the future.

REFERENCES

[1] Eytan Adar, Jaime Teevan, and Susan T Dumais. 2008. Large scale analysis of web revisitation patterns. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 1197–1206.

[2] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. 2014. The dynamics of repeat consumption. In *Proceedings of the 23rd international conference on World wide web*. 419–430.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.0473

[4] Austin R Benson, Ravi Kumar, and Andrew Tomkins. 2016. Modeling user consumption sequences. In *Proceedings of the 25th International Conference on World Wide Web*. 519–529.

[5] Rahul Bhagat, Srevatsan Muralidharan, Alex Lobzhanidze, and Shankar Vishwanath. 2018. Buy it again: Modeling repeat purchase recommendations. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 62–70.

[6] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *CoRR* abs/2010.03240 (2020). arXiv:2010.03240 https://arxiv.org/abs/2010.03240

[7] Jun Chen, Chaokun Wang, and Jianmin Wang. 2015. Will you" reconsume" the near past? fast prediction on short-term reconsumption behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[8] Jun Chen, Chaokun Wang, Jianmin Wang, and S Yu Philip. 2016. Recommendation for repeat consumption from user implicit feedback. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (2016), 3083–3097.

[9] Junsu Cho, SeongKu Kang, Dongmin Hyun, and Hwanjo Yu. 2021. Unsupervised Proxy Selection for Session-based Recommender Systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 327–336.

[10] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[11] Minjin Choi, Jinhong Kim, Joonseok Lee, Hyunjung Shim, and Jongwuk Lee. 2021. Session-aware linear item-item models for session-based recommendation. In *Proceedings of the Web Conference 2021*. 2186–2197.

[12] Michael D Ekstrand and Daniel Kluver. 2021. Exploring author gender in book rating and recommendation. *User modeling and user-adapted interaction* 31, 3 (2021), 377–420.

[13] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. Sequence and time aware neighborhood for session-based recommendations: Stan. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1069–1072.

[14] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06939

[16] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-Rich Session-Based Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 241–248. https://doi.org/10.1145/2959100.2959167

[17] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 306–310.

[18] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[19] Dimitrios Kotzias, Moshe Lichman, and Padhraic Smyth. 2018. Predicting consumption patterns with repeated and novel events. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2018), 371–384.

[20] Lukas Lerche, Dietmar Jannach, and Malte Ludewig. 2016. On the value of reminders within e-commerce recommendations. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. 27–35.

[21] Fenghuan Li, Zhaosheng Lin, and Zhenyu Wang. 2018. Tag-Driven Online Novel Recommendation with Collaborative Item Modeling. *Information* 9, 4 (2018). https://doi.org/10.3390/info9040077

[22] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[23] Jie Liu, Chun Yu, Wenchang Xu, and Yuanchun Shi. 2012. Clustering web pages to facilitate revisitation on mobile devices. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. 249–252.

[24] Tie-Yan Liu. 2011. Learning to rank for information retrieval. (2011).

[25] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction* 28, 4 (2018), 331–390.

[26] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2021. Empirical analysis of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction* 31, 1 (2021), 149–181.

[27] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 43–52. https://doi.org/10.1145/2766462.2767755

[28] Jérémie Rappaz, Julian McAuley, and Karl Aberer. 2021. Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption. In *Fifteenth ACM Conference on Recommender Systems*. 390–399.

[29] Markus Reiter-Haas, Emilia Parada-Cabaleiro, Markus Schedl, Elham Motamedi, Marko Tkalcic, and Elisabeth Lex. 2021. Predicting music relistening behavior using the ACT-R framework. In *Fifteenth ACM Conference on Recommender Systems*. 702–707.

[30] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4806–4813.

[31] Jaime Teevan, Eytan Adar, Rosie Jones, and Michael Potts. 2006. History repeats itself: repeat queries in Yahoo's logs. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 703–704.

[32] Sarah K Tyler and Jaime Teevan. 2010. Large scale query log analysis of re-finding. In *Proceedings of the third ACM international conference on Web search and data mining*. 191–200.

[33] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems* 28 (2015).

[34] Mengting Wan and Julian McAuley. 2018. Item Recommendation on Monotonic Behavior Chains. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 86–94. https://doi.org/10.1145/3240323.3240369

[35] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *The World Wide Web Conference*. 1977–1987.

[36] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Defu Lian. 2021. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)* 54, 7 (2021), 1–38.

[37] Bo Zhao, Koichiro Narita, Burkay Orten, and John Egan. 2018. Notification Volume Control and Optimization System at Pinterest. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1012–1020. https://doi.org/10.1145/3219819.3219906

[38] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving Recommendation Lists through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web* (Chiba, Japan) *(WWW '05)*. Association for Computing Machinery, New York, NY, USA, 22–32. https://doi.org/10.1145/1060745.1060754