DEIM Forum 2022 H24-1

Natural Language Generation Enhanced Recommender System

Yi YU† and Kazunari SUGIYAMA†

† Graduate School of Informatics, Kyoto University
Yoshida-honmach, Sakyo-ku, Kyoto, 606–8501, Japan
E-mail: yuyi@db.soc.i.kyoto-u.ac.jp,kaz.sugiyama@i.kyoto-u.ac.jp

Abstract In recent years, a large number of explainable recommendation models have been proposed, aiming at generating not only high-quality recommendations but also intuitive explanations. However, current mainstream solutions are either to retrieve-and-rank sentences or pre-defined templates, which limits the diversity and expressiveness of the generated explanations. In this work, we enhance the explainable recommender systems by using Transformer architecture [25] to generate natural language explanations via a multi-task learning framework. In the encoding stage, we first encode the users and items into latent space and learn the context representation by predicting rating scores and reviews properties (e.g., sentiment polarity, recency, length). For explanation generation, Transformer Decoder blocks are employed to decode the context representation into a natural sentence.

Key words Natural Language Generation; Recommender System; Transformer

1 Introduction

Recent years have witnessed a continuous boom in the field of recommender systems research. Recommender systems improve users experience in online services in a variety of areas, like e-commerce applications, social networking service, media platforms and many others, by providing personalized suggestions and recommendations. Traditional recommender systems such as collaborative filtering [3, 11, 19] and review-based methods [18, 2, 33, 30], predict only users' preference over a list of items, where highly relevant items are predicted to score higher (or to rank higher in terms of ranking problem setting). However, focusing solely on the accuracy metrics for evaluating recommendation algorithms results in more and more complicated design of the recommender systems. For this reason, current recommendation models are inevitably becoming unexplainable and less transparent to both designers and users. To address this problem, explainable recommender systems have received much attention [31].

The task of explainable recommendation is not only to predict a score estimating how likely the target user prefers the target item, but also to provide an explanation to justify why the item should be recommended. For instance, given a pair of user ID u and item ID i, an explainable recommender system predicts a rating $\hat{r}$ 4 out of 5, as well as a justification like *"As you value the quality of service, we recommend this restaurant to you"*. Regarding the justification, it may have various representations. For example, explanation could be textual (e.g., words, tips, sentences), or visual explanation (e.g., highlighted parts of an image, radar chart). In our work, we focus only on the text data, so the "justification" or "explanation" in the following work means textual sentence. An example of explainable recommendation is shown as Table 1, where the explanation column is extracted from the user-reviews data[!ℭm1!K].

Rating	**Explanations**
5	Great story! Everyone needs a pocket.
5	Great little children's story.
3	Nakedness not necessary!
2	Disappointed the episode was included with DVD.
5	Better than I expected given the price.
5	This is a very nice calculator for high school students in higher math.
3	Too much error code...does not work for me and gives a lot of problems.
4	This is a really good, high-quality product.
5	It is highly recommended for students from high school on.

Table 1 : Examples of the ratings and ground truth explanations from Amazon Review Data.

Most existing methods [10, 32, 5, 26, 20, 14, 21, 13] for explainable recommendation apply text processing technology including topic modeling and sentiment analysis and so on, on user-reviews for extracting users' points of interest and the items' inherent characteristics (i.e.,features). However, these methods have some inherent limitations: (a) The constructed

[!ℭm1!K]'https://jmcauley.ucsd.edu/data/amazon/

explanations are either retrieved sentences from candidates, or pre-defined sentences with extracted features. Hence, the generated sentences are limited in expressiveness and diversity; (b) A large proportion of user-reviews are noisy or irrelevant sentences, resulting in poor performance in both extraction and generation phases. For instance, extremely long reviews and extremely short reviews (e.g., "good") may be useless resources; old reviews may be less useful than recent reviews, and so on.

To address the limitation above, we propose to model the reviews with their associated available properties, as inspired by [27], and employ Transformer [25] neural network architecture for generating free-sentence explanations. In the work [27], they compute property scores from different aspects and then bridge the relationship between review properties with review usefulness for deriving property attention into the network, in which the objective function is a single task, i.e, rating prediction. In contrast, we employ the review properties for designing a novel prediction task, aiming at deriving a **context embedding** for target user and target item. The context embedding will then be combined with originally learned user and item embeddings with the rating prediction task, for generating the explanation (i.e., word sequence). Therefore, we propose a novel model, **TERRP**, representing **T**ransformer-based **E**xplainable **R**ecommendation via **R**eview **P**roperty. TERRP is multi-task learning framework, consisting of three modules, i.e., rating regression block, property encoding block, and explanation generation block. For rating regression, a multi-layer perceptron network (MLP) is used to project user embeddings and item embeddings into ratings. For property encoding block, we use Transformer Encoder and a Softmax layer to project the user and item embeddings into property space, which is a quadruple $(Rencency, Length, HelpfulVotes, SentPolarity)$ [!JCm2!K]. We then extract the last layer output as context embedding. For explanation generation, we adopt Transformer Decoder to decode the concatenation of user embedding, item embedding, and the context embedding into a sequence of words, which is exactly the final explanation. All the parameters in the three modules are learned by a multi-task learning approach in an end-to-end training manner.

Given these points, the main contributions of this paper are summarized as follows:

- We propose a novel explainable recommender algorithm, TERRP, which can generate free-text sentences explaining why item matches user' interest, in addition to the prediction of rating preference.
- TERRP leverages user-review properties as well as textual information. The usage of property encoding module based on Transformer architecture can enhance the explanation generation performance.
- We also design a embedding alignment mechanism in rating regression module, which can improve the performance of recommendation.

2 Related Work

2. 1 Recommendation with Review-properties

Collaborative filtering has been studied for a long time and achieve some success in recommendation systems [24, 22], which takes the user-item rating matrix as the input. Those approaches' performance would decrease when encountering sparse matrices. Therefore, some works consider incorporating the text information with the rating matrix to improve the prediction performance. For example, [10] integrates topic modeling technique based on user-reviews to generate the latent factors for both users and items, and then predict the ratings by using the learned latent representations. [16] introduce a multi-task learning framework based on gated recurrent unit (GRU). In training stage, the model iteratively updates user and item embeddings by jointly learning from the loss of rating prediction task, review text generation task, and review tips generation task. The method can also generate tips (a short sentence) as explanations for the recommendation. Recently, in [27], the authors propose to use review properties (e.g., length, age, sentiment polarity) for modeling the usefulness of the reviews. For learning the latent presentation of users and items, they adopt convolutional neural network (CNN) combined with attention mechanism for capturing the information7. Although their work explores the effectiveness of using reviews' attributes, they cannot generate either justification or explanations, i.e., not explainable recommendation.

2. 2 Explainable Recommender Systems

Explainable Recommendation has been studied from two major perspectives: human-computer interaction and machine learning [32, 31]. The former [9, 5] investigates how people perceive different styles of explanations, while the latter provides explanations by designing new explainable recommendation algorithms. There exist various types of explanation styles, such as pre-defined templates [32, 12] , ranked sentences [6, 13], image visualizations [7], knowledge graph paths [29, 8, 28] , reasoning rules [23, 4, 34], and so on. Recently, generated natural language explanations [20, 14, 15] have received much attention, mainly owing to the advancement of natural language generation technology and the availability of textual data on recommendation

[!JCm2!K] We will elaborate on this in Section 3.

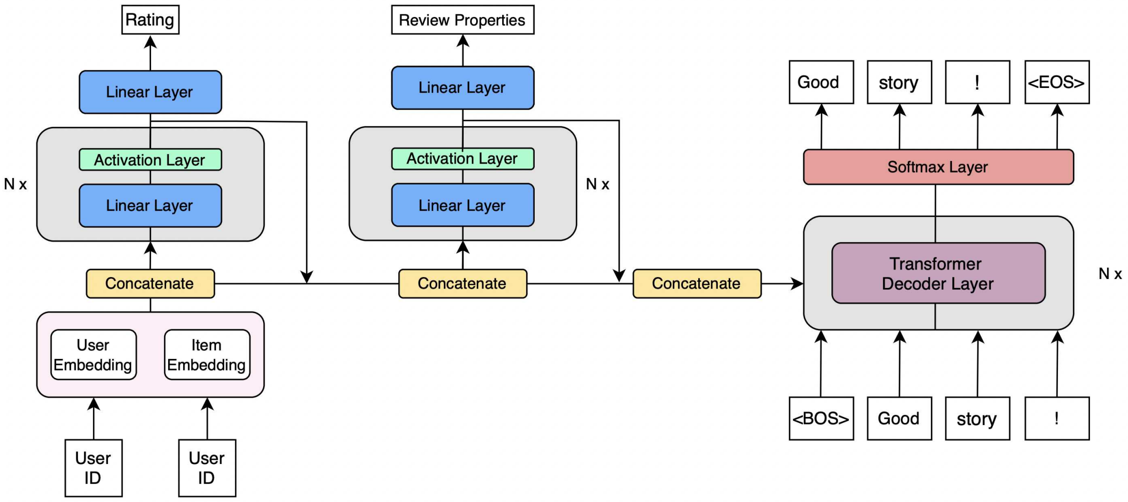

Figure 1 The framework of our proposed model TERRP.

platforms such as e-commerce. However, they ignore the inherent noises of, the attributes of, and the usefulness of the review data, resulting in the inevitably low-quality of the constructed free sentences, which motivates this work.

3 Problem Formulation

For each user u (or each item i), we have a review set $\mathbf{R^u} = \{R_1^u, R_2^u, ..., R_l^u\}$ consisting all reviews written by u (or $\mathbf{R^i}$ written for i), where R_i^u represents one record which essentially is a sequence of words. We then extract properties based on each record and thus obtain the property set $\mathbf{P} = \{P_1, P_2, ..., P_l\}$, where each element P_i corresponds to each review R_i (we omit the superscript u or i in the following and use R instead of R^u, R^i for brevity). P_i is a quadruple $(Rencency, Length, HelpfulVotes, SentPolarity)$ representing the number of days since the user's commented, the word count of the review, the number of votes by other users who found the comment useful, and sentiment polarity of the review (1 for positive and -1 for negative), respectively.

Given a user u and an item i, as well as their reviews sets $\mathbf{R^u}, \mathbf{R^i}$, our target is to generate a rating $\hat{r}_{u,i}$ that predict u' preference toward i, as well as a natural language sentence $\hat{S}_{u,i}$ that would explain why item i fits user u' interests.

4 Framework

In this section, we introduce our proposed TERRP model to learn user and item representations from reviews and properties for explainable recommendation. The architecture of our approach is shown in Figure 1. Comment it out first, for compile faster.

It has three modules standing for the three different tasks: rating prediction task, review properties prediction task, and explanation generation task. Regarding the rating prediction, we employ multiple linear layer in addition to non-linear activation layer on user and item latent factors. After learning the representations from the rating information, we then concatenate the embeddings and last hidden vector as a starting point for review properties prediction, which are quadruple $(Rencency, Length, HelpfulVotes, SentPolarity) \in \mathbb{R}^4$. We choose review properties prediction as auxiliary task because it bridges the rating prediction task and explanation generation task closely. More specifically, user's rating on item directly reflects whether he/she is positive or negative about the item. Thus, the embeddings learned from ratings prediction task could be further improved by review sentiment polarity prediction. In addition, the property encoding of review could help determine the how helpful the review is in generating explanations. Therefore, the Transformer Decoder pays different attentions on the user-item reviews.

4.1 Rating Prediction

In recommendations, the task is to estimate a rating that a user will give to an item. This matrix completion problem also can be viewed as a generalization of classification or regression problem [1]. The main difference is that traditional classification task has fixed position and number of features and targets, whereas rating prediction task has unfixed columns and unknown number of targets (the observations vary from user to user). Latent factor-based models are widely used approach for this task, in which users and items are modeled as low-dimensional representations.

Given the IDs of user and item, We first use an Embedding Layer (a simple look up table that stores embeddings of a fixed dictionary and size) to obtain initial latent representations of user and item denoted as $u \in \mathbb{R}^{k_u}$ and $v \in \mathbb{R}^{k_v}$, respectively.

We then employ multiple perceptron (MLP) layers to map the embeddings into a rating score and the hidden output denoted as $\hat{r}$, h_l as follows:

$$\begin{cases} h_1^r = ReLU(\mathbf{W_1}^r \cdot Concat[u, v] + \mathbf{b}_1^r) \\ h_2^r = ReLU(\mathbf{W_2}^r \cdot h_1^r + \mathbf{b}_2^r) \\ ... \\ h_l^r = ReLU(\mathbf{W}_l^r \cdot h_{l-1}^r + \mathbf{b}_l^r) \\ \hat{r} = \mathbf{W}_{l+1}^r \cdot h_l^r + \mathbf{b}_{l+1}^r \end{cases} \quad (1)$$

, where $\mathbf{W}^r$ and $\mathbf{b}^r$ are learnable weights and bias of the MLP layers, h_i^r are hidden vectors, and ReLU is the non-linear activation function.

The goal of rating prediction is to:

$$Minimize\,\mathcal{L}_r : \mathcal{L}_r = \sum_{u \in U, i \in I} (\hat{r}_{u,i} - r_{u,i}) \quad (2)$$

For rating prediction, we train the model solely on the rating matrix. After training, the updated user and item embeddings has probably learned useful information (e.g., user's personalized preference towards the item) from the ratings distributions. We lock the trained user and item latent embeddings denoted as $[u^r, v^r]$, and in addition make a copy of them for future training. We retain the last hidden layer output h_l^r for further tasks.

4. 2 Property Encoding

To generate the sentences that explain why a item fits or not fits user's interest, it is insufficient to use only the user and item embeddings learned from rating prediction as hidden state input. The reason is that the learned latent factors may to some extent show user's sentiment polarity towards the item (e.g. 5 stands strong positive, 1 stands strong negative), but it has little information specified to user's aspects and item's properties (e.g., price, features). Therefore, learning from reviews may help to achieve this goal. Suspecting that not all reviews are equally useful at contributing to such information, we design a novel task, review property prediction, to enhance the explanation generation learning.

We represent each review by its property quadruple $(Rencency, Length, HelpfulVotes, SentPolarity) \in \mathbb{R}^4$. The goal is to use user and item embeddings $[u^r, v^r]$ as well as hidden state h_l^r to predict the property quadruple of reviews. We use n-layers MLP for this task as follows:

$$\begin{cases} h_1^p = ReLU(\mathbf{W_1}^p \cdot Concat([[u^r, v^r], h_l^r], +\mathbf{b}_1^p) \\ h_2^p = ReLU(\mathbf{W_2}^p \cdot h_1^p + \mathbf{b}_2^p) \\ ... \\ h_l^p = ReLU(\mathbf{W}_l^p \cdot h_{l-1}^p + \mathbf{b}_l^p) \\ \hat{p} = \mathbf{W}_{l+1}^p \cdot h_l^p + \mathbf{b}_{l+1}^p \end{cases} \quad (3)$$

, where $\mathbf{W}^p$ and $\mathbf{b}^p$ are learnable weights and bias for property prediction task, h_i^p are hidden vectors, and ReLU is the non-linear activation function.

Note that property quadruple has four elements with different scales (e.g., *Rencency* in scale $[0, 1000+]$, whereas *SentPolarity* in scale $[-1, 1]$), thus it is needed for normalizing them into the same scale. We employ Min-Max Normalization for this purpose as follows:

$$x' = \frac{x - min(x)}{max(x) - min(x)}. \quad (4)$$

The goal of property prediction is to:

$$Minimize\,\mathcal{L}_p : \mathcal{L}_p = \sum_{u \in U, v \in I} \sum_{i \in p} (\hat{p}_{u,v}^i - p_{u,v}^i) \quad (5)$$

After training on the reviews, the updated user and item embeddings has been further updated, denoted as $[u^p, v^p]$. We then lock the embeddings denoted as $[u^p, v^p]$, and in addition make a copy of them for future training. We retain the last hidden layer output h_l^p for further tasks.

4. 3 Explanation Generation

We concatenate the two user and item representations, i.e., u^r, v^r and u^p, v^p that are learned from rating prediction and property prediction task, respectively. We also concatenate the two hidden layer outputs h_l^r and h_l^p. Therefore, we use the resulting vector $h = Concat([[u^r, i^r, v^p, v^p], [h_l^r, h_l^p]])$ as the initial hidden state for Transformer Decoder input. However, as the dimension of h is huge due to the concatenation, to reduce the number of parameters, we first pass it to a linear transformation into a relatively low-dimension space.

Then, we input the context hidden state h and <BOS> token vector (i.e., Begin of Sentence) $x_{<bos>}$ into Transformer decoder and generate the final sequence of words. The objective is to minimize the difference between the generated sequence s and ground-truth sentence. We use the Negative Log-Likelihood (NLL) as the loss function as follows:

$$\mathcal{L}_e = -\sum_{c \in s}^{s} \sum_{k=1}^{|\mathcal{V}|} \mathbf{c}^{(k)} \log \hat{\mathbf{c}}^{(k)}, \quad (6)$$

At the testing stage, given a trained model, we use the beam search to find the best sequence s having the maximum log-likelihood:

$$s^* = \arg\max_{s \in \mathcal{S}} \sum_{w \in s} \log \hat{\mathbf{s}}, \quad (7)$$

4. 4 Multi-task Learning

We integrate all the sub-tasks of rating prediction, property encoding and explanation generation into a unified multi-task learning framework whose objective function is:

$$\mathcal{L}_{total} = \min_{\Theta} \left(\lambda_r \mathcal{L}_r + \lambda_p \mathcal{L}_p + \lambda_s \mathcal{L}_e \right), \quad (8)$$

where Θ denotes the trainable parameters in the whole framework, and λ_r, λ_p and λ_e are regularization weights that balance the learning of different tasks.

4.5 Inference

In the inference stage, we have only the user ID and item ID for both predicting the rating and constructing explanation sentence. Note that after training, we have already kept three groups of user and item embeddings, i.e., U^r, V^r, U^p, V^p and U^e, V^e.

For rating prediction, we use the embeddings learned specifically for the first task, U^r, V^r, and pass the input user ID and item ID into TERRP model. The output is a scale value in range 1 to 5, representing the preference of user towards the item.

For explanation generation, we first load the embeddings U^p, V^p and predict the review properties first; then we reload the embeddings U^e, V^e for generating the explanation, in which we greedily pick up the token with highest probability for the next time step generation.

5 Conclusions

Summary. In this paper, we propose a Transformer-based explainable recommendation model, TERRP, to simultaneously predict precise ratings and construct free-text yet relevant justifications. TERRP leverages user-review properties (such as length, recency, number of helpful votes) as well as textual information of user-reviews. The usage of property encoding module based on Transformer architecture can greatly boost the generation of explanation sentence. In addition, in the rating prediction module, we also design a embedding alignment mechanism that aligns user and item embedding so as to enhance the performance of recommendation. We believe that explainable recommender systems could go further with the help of natural language processing, in which free-text generation is a promising solution.

Future Work. We will implement extensive experiments on large datasets to prove the effectiveness and superiority of our method TERRP. Furthermore, we will explore other possibilities on the joint of recommendation with texts data for explanation recommendation. For example, pre-training and prompting learning methods may be promising solutions to our case, as they have been demonstrated superior in other domains of natural language processing [17].

References

[1] Charu C Aggarwal et al. *Recommender systems.* Vol. 1. Springer, 2016.

[2] Amjad Almahairi et al. “Learning Distributed Representations from Reviews for Collaborative Filtering”. In: *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15).* 2015, pp. 147–154.

[3] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. “Matrix Factorization Techniques for Context Aware Recommendation”. In: *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys ’11).* 2011, pp. 301–304.

[4] Hanxiong Chen et al. “Neural Collaborative Reasoning”. In: *Proceedings of the Web Conference 2021 (WWW ’21).* 2021, pp. 1516–1527.

[5] Li Chen and Feng Wang. “Explaining Recommendations based on Feature Sentiments in Product Reviews”. In: *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI 2017).* 2017, pp. 17–28.

[6] Xu Chen, Yongfeng Zhang, and Zheng Qin. “Dynamic Explainable Recommendation based on Neural Attentive Models”. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-19).* Vol. 33. 01. 2019, pp. 53–60.

[7] Xu Chen et al. “Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’19).* 2019, pp. 765–774.

[8] Zuohui Fu et al. “Fairness-aware Explainable Recommendation over Knowledge Graphs”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’20).* 2020, pp. 69–78.

[9] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. “How Should I Explain? A Comparison of Different Explanation Types for Recommender Systems”. In: *International Journal of Human-Computer Studies* 72.4 (2014), pp. 367–382.

[10] Xiangnan He et al. “Trirank: Review-aware Explainable Recommendation by Modeling Aspects”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM ’15).* 2015, pp. 1661–1670.

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. “Matrix Factorization Techniques for Recommender Systems”. In: *Computer* 42.8 (2009), pp. 30–37.

[12] Lei Li, Li Chen, and Ruihai Dong. "CAESAR: Context-Aware Explanation based on Supervised Attention for Service Recommendations". In: *Journal of Intelligent Information Systems* 57.1 (2021), pp. 147–170.

[13] Lei Li, Yongfeng Zhang, and Li Chen. "EXTRA: Explanation Ranking Datasets for Explainable Recommendation". In: *The 44th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '21)*. 2021, pp. 2463–2469.

[14] Lei Li, Yongfeng Zhang, and Li Chen. "Generate Neural Template Explanations for Recommendation". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*. 2020, pp. 755–764.

[15] Lei Li, Yongfeng Zhang, and Li Chen. "Personalized Transformer for Explainable Recommendation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*. 2021, pp. 4947–4957.

[16] Piji Li et al. "Neural Rating Regression with Abstractive Tips Generation for Recommendation". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. 2017, pp. 345–354.

[17] Pengfei Liu et al. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *arXiv Preprint arXiv:2107.13586* (2021).

[18] Julian McAuley and Jure Leskovec. "Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text". In: *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. 2013, pp. 165–172.

[19] Andriy Mnih and Russ R Salakhutdinov. "Probabilistic Matrix Factorization". In: *Advances in Neural Information Processing Systems (NIPS 2008)*. 2008, pp. 1257–1264.

[20] Jianmo Ni, Jiacheng Li, and Julian McAuley. "Justifying Recommendations Using Distantly-labeled Reviews and Fine-grained Aspects". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*. 2019, pp. 188–197.

[21] Reinald Adrian Pugoy and Hung-Yu Kao. "Unsupervised Extractive Summarization-Based Representations for Accurate and Explainable Collaborative Filtering". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*. 2021, pp. 2981–2990.

[22] Francesco Ricci, Lior Rokach, and Bracha Shapira. "Introduction to Recommender Systems Jandbook". In: *Recommender Systems Handbook*. Springer, 2011, pp. 1–35.

[23] Shaoyun Shi et al. "Neural Logic Reasoning". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*. 2020, pp. 1365–1374.

[24] Xiaoyuan Su and Taghi M Khoshgoftaar. "A Survey of Collaborative Filtering Techniques". In: *Advances in Artificial Intelligence* 2009 (2009).

[25] Ashish Vaswani et al. "Attention is All You Need". In: *Advances in Neural Information Processing Systems (NIPS 2017)*. 2017, pp. 5998–6008.

[26] Nan Wang et al. "Explainable Recommendation via Multi-task Learning in Opinionated Text Data". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 2018, pp. 165–174.

[27] Xi Wang, Iadh Ounis, and Craig Macdonald. "Leveraging Review Properties for Effective Recommendation". In: *Proceedings of the Web Conference 2021 (WWW '21)*. 2021, pp. 2209–2219.

[28] Yikun Xian et al. "CAFE: Coarse-to-Fine Neural Symbolic Reasoning for Explainable Recommendation". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*. 2020, pp. 1645–1654.

[29] Yikun Xian et al. "Reinforcement Knowledge Graph Reasoning for Explainable Recommendation". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in information retrieval (SIGIR '19)*. 2019, pp. 285–294.

[30] Yinqing Xu, Wai Lam, and Tianyi Lin. "Collaborative Filtering Incorporating Review Text and Co-clusters of Hidden User Communities and Item groups". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14)*. 2014, pp. 251–260.

[31] Yongfeng Zhang and Xu Chen. “Explainable recommendation: A survey and new perspectives”. In: *arXiv preprint arXiv:1804.11192* (2020).

[32] Yongfeng Zhang et al. “Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis”. In: *Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR ’14)*. 2014, pp. 83–92.

[33] Lei Zheng, Vahid Noroozi, and Philip S Yu. “Joint Deep Modeling of Users and Items Using Reviews for Recommendation”. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM ’17)*. 2017, pp. 425–434.

[34] Yaxin Zhu et al. “Faithfully Explainable Recommendation via Neural Logic Reasoning”. In: *Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL ’21)*. 2021, pp. 3083–3090.