Integrating Social Environment in Machine Learning Model for Debiased Recommendation

Yihong Zhang[1](✉), Lina Yao[2], and Takahiro Hara[1]

[1] Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
yhzhang7@gmail.com, hara@ist.osaka-u.ac.jp

[2] School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia
lina.yao@unsw.edu.au

Abstract. Social data provided by social media platforms contains rich social environment information, which has been used in many ubiquitous tasks such as disaster monitoring, epidemic tracking, or stock market movement prediction. In this paper, we show that social environment information can be used to debias recommendation system. Recommendation systems are used to extract the preference of e-commerce platform users for predicting what would they see or buy next. Most recommendation systems that rely on machine learning models are currently facing the problem of bias, which occurs when the system is isolated in a platform and trained solely on past data. While various debiasing methods have been proposed, the problem remains largely unsolved. In response, we propose a recommendation model that uses real-time social data to reduce recommendation bias while also improving accuracy. The proposed model integrates social data, represented as embeddings generated by language models, into a traditional machine learning model called single value decomposition (SVD). Empirical evaluations on two pairs of real-world e-commerce plus social data datasets show that our model is superior in both recommendation accuracy and bias reduction compared to state-of-the-art debiased recommendation methods.

Keywords: environmental factor extraction · social media mining · debiased recommendation system

1 Introduction

Social media platforms facilitate user-generated content, outputting hundreds of millions of text messages and other data every day, covering almost any topic in social conversation [27]. These social data can be seen as a representation of the

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 594, pp. 219–230, 2024.
https://doi.org/10.1007/978-3-031-63992-0_14

whole social environment and have been used to help many ubiquitous computational tasks [17]. For example, it has been shown that tweets from Twitter can be used for detecting earthquakes [24] and monitoring epidemic spreads [2]. In some tasks, the environmental factor has been extracted from social data to improve an existing model. Example of these tasks include crime prediction, product sales forecasting, stock market movement prediction, and traffic monitoring.

Recommendation systems have now been widely used in many internet platforms, such as Amazon, eBay, and YouTube, providing personalized information to hundreds of millions of users worldwide [9]. Currently, recommendation system research is facing a critical situation that the system, consisting of a recommendation model, data, and users, is usually isolated in a specific platform. As a result, recommendation systems are developing so-called *bias*, due to the fact that the model is trained only on the past, static data [8]. Even though many different recommendation models have been proposed, bias is prevalent in almost all models [15]. An example impact of the bias is that the item popular in the past will get more recommendations, and will become even more popular in the future, while unpopular items get fewer recommendations and become even more unpopular. This is called *popularity bias* [1], one of the known biases in an isolated recommendation system, and is known as being connected to fairness and diversity.

We argue that environmental information extracted from social data can also be helpful in improving the recommendation of e-commerce products. For example, a festival discussed on social media can invoke wine purchases [31]. A natural disaster that is happening and discussed on social media can be followed by emergency goods purchases. Furthermore, seasonal information discussed on social data can also be hints for selling certain products [28]. Such social context may not be found within the recommendation system, but is present in social data. The problem is how to incorporate useful social data into existing recommendation systems.

Our insight of studying recommendation bias is that one of the causes of such bias is the fact that most recommendation systems are built on a machine learning model, which is isolated and trained only on past data within the system. This causes *feedback loop* which leads to bias. As such, if we can bring new information and out-of-system contexts to such an isolated system, illustrated in Fig. 1, we can expect less bias and even better recommendation accuracy. We use social data from social media platforms as the new context because they contain the latest information and can be easily obtained. A challenge, though, is what data should we use and extract from the massive corpus.

Social data are largely unstructured and noisy. It has been a challenging problem to represent real-time social data [11]. The emergence of pre-trained language models [3], however, gives us a convenient means for this task. Pre-trained language models such as GPT [12] and BART [20] let us input unstructured text and obtain embeddings that contain subtle information. In this paper we use a pre-trained language model to obtain social data embeddings, and then integrated them as the environmental factor to the well-known single value decomposition

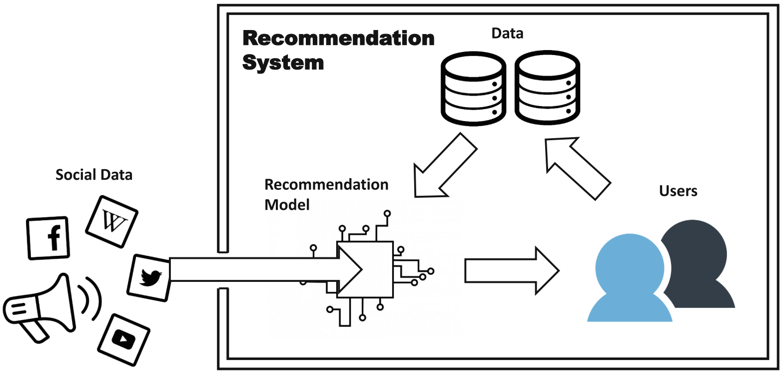


Fig. 1. Enhancing isolated recommendation system with social data

(SVD) learning model, similar to other latent factor models [18,19]. Testing on two pairs of real-world e-commerce and social data datasets, we show that our model is superior in terms of both the recommendation accuracy and the bias reduction, compared to state-of-the-art debiased recommendation methods. To summarize, our main contributions with this paper are:

- We identify the problem of recommendation bias as the result of an isolating learning process. In contrast to other hypotheses, our direction is to reduce the isolating state of the learning model.
- We propose to use the environmental factor extracted from social data to reduce the isolated state of recommendation models and improve the recommendation diversity. To the best of our knowledge, this is the first work that proposes to use real-time social data to reduce biases in a recommendation system.
- We conduct comprehensive empirical evaluations on two pairs of e-commerce and social media datasets. The evaluation results show that our approach not only reduces the bias, but improves recommendation accuracy in general.

2 Related Work

Some research efforts have already explored methods of using social data to enhance the recommendation system of another domain [7,10]. However, many of these methods have strict assumptions, such as shared users between the social media platform and the recommendation domain, or famous items that can be linked to entities in social data [26,33]. Such assumption significantly limits their applicability. In our work, we do not assume any shared data between the social media platform and the recommendation domain. The social data in our work represents the whole social environment, from which the recommendation system can borrow information as needed. In this way, we do not require the item to be famous due to the coverage and the details of the social data.

Our work also takes inspiration from recommendation debiasing works. For example, Zhang et al. proposed that the popularity bias is partially caused by the temporary item bias [30]. They used a deconfounding method to reduce the bias. Misztal-Radecka and Indurkhya proposed to analyze recommendation disparity for detected biased attributes [22]. Following the finds, they proposed a debiased model selection method. Chaney et al. investigated how algorithmic confounding in recommendation systems increases homogeneity and decreases utility [6]. These studies, however, restrain themselves within the system, which makes their effectiveness limited.

3 Debiasing Recommendation with Social Data

We deal with the traditional implicit recommendation problem, with the exception that we allow the addition of out-of-system social data. In this section, we will first define the problem. Then we will present our method for integrating the real-time massive social data to a recommendation system.

3.1 Problem Formulation

The base problem we are dealing with is making dynamic personalized recommendations according to user preference and time. We follow the approach of collaborative filtering with implicit feedback. In a dynamic setting, the model should make a time-aware prediction $\hat{y}_{uit} = g(u, i, t)$, where $\hat{y}_{uit}$ is the predicted preference of user u towards item i at time t.

The main point of this paper is to incorporate social data into a dynamic recommendation model to improve recommendation performance. We assume we are monitoring social media in real-time, so that we have the data for the current timeframe and some past timeframes. As such, for each timeframe t, we have a collection of short text documents D_t. This collection of documents contains rich data. To use them in computational tasks, however, there may be two issues. First, the volume of data is large, with each D_t potentially containing millions of short texts. Second, the length of text documents in D_t is different. This will cause a problem for computational models that rely on fixed-length inputs. To address the issues, we need to have a transformation function

$$R(t) = \rho(D_t) \tag{1}$$

to transform the data into a representation that is much smaller than the data and has a fixed length for each timeframe t, while still preserving the most important information for the timeframe.

To use social data to support the model is to incorporate the social data representation R as an extra input to the model, so that the prediction becomes

$$\hat{y}_{uti} = g(u, i, t, R(t)). \tag{2}$$

Our task, then, is to devise the social data representation R and the integrated recommendation model g.

3.2 Social Data Representation

Traditional text stream techniques that can be used to represent social data include bag-of-words and topic modeling [4]. One common deficiency of these techniques is that they treat words as independent entities, not considering the sequence or context in the text. Natural language processing (NLP) research has since then produced techniques such as BERT [29] that can represent text while taking into account the context. Recently, pre-trained language models such as GPT [5] have shown superiority in many NLP tasks such as text summarization and question answering. One reason they have this superiority is due to their highly sophisticated text representation. In this paper, we propose to take advantage of the superior text representation capability of a pre-trained language model to help our task. Specifically, we use a language model to provide a mapping from text to real-value embeddings:

$$E(t) = \mathrm{LM}(D_t) \tag{3}$$

There are several choices for the pre-trained language models. In this paper we use BART [20] due to its efficiency when running on local computers. We use the *facebook/bart-base* model[1], which is a median size language model with 139M parameters. This can be run on our local computer with one NVIDIA GeForce GTX 1070 GPU with 8 GB graph memory. We can expect that larger and more complex models may provide better results. But these larger models are either too slow to run locally or too expensive to run as an online service, as we have potentially millions of text to process. Thus given our experimental environment, the BART model is a suitable choice.

Furthermore, we can calculate embeddings for a smaller time unit, such as an hour, and stack them together for a larger time unit, such as a day:

$$R(t) = \{E(t_s) : \forall t_s \in t\} \tag{4}$$

In this way, we essentially produce for each timeframe t a matrix of $n \times m$, where n is the number of small time units in a larger time unit, e.g., 24 in the case of hours in a day, and m is the text embedding size.

3.3 Recommendation Model

Our base recommendation model is a latent factor model [18]. In a latent factor model, users and items have corresponding latent factor embeddings, denoted as p_u and q_i. When we combine a pair of user-item embeddings through multiplication, i.e., $p_u \cdot q_i$, we can reveal user u's preference towards item i. And there are global latent factors for each user and item, denoted as b_u and b_i, also called user and item biases. The complete latent factor model is thus

$$\hat{y}_{uit} = p_u \cdot q_i + b_i + b_u, \tag{5}$$

[1] https://huggingface.co/facebook/bart-base.

where the parameters can be learned through gradient descent techniques. We note that in a complete implementation, traditional regularizers are to be used, but are not shown here for simplicity.

As we mentioned in the introduction, an e-commerce item may suddenly become important due to some social media discussions at the time. As such, on the base recommendation model, we would like to add a social factor, denoted as $s_{i,t}$, that indicates the item's significance with regard to the social media discussion at the time. So the extended model becomes:

$$\hat{y}_{uit} = p_u \cdot q_i + b_i + b_u + s_{i,t}. \tag{6}$$

We call the model Language Model-enhanced Social Recommender (LMSocRec). Now we deal with how to generate $s_{i,t}$ given a social media representation $R(t)$. We need a method to make this latent factor cover both the social data and the item characteristics, and we find that the *attention* [25] is suitable for this task. Following the approach of attention, we can make a social context encoder for each item at a time. The module is depicted in Fig. 2.

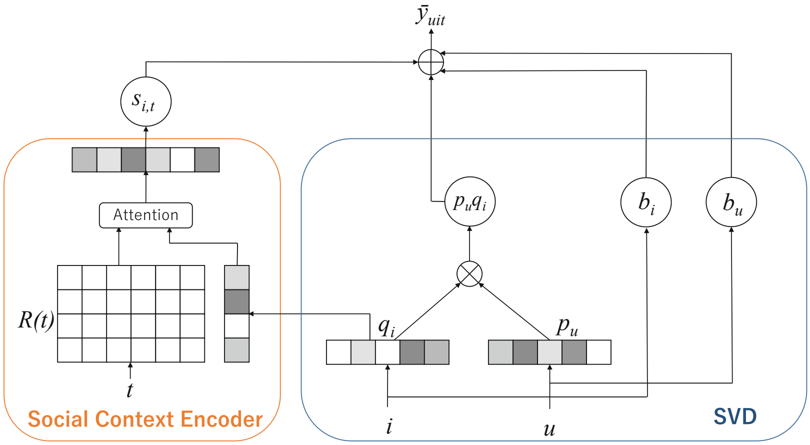


Fig. 2. Overview of the LMSocRec model

Since we assume no shared information between social data and the recommendation system, the item embedding size may be different from dimensions of $R(t)$. We use a linear transformation on the item embedding:

$$r_i = \sigma(\mathbf{h}_i^\top \cdot q_i) \tag{7}$$

where σ is a linear transformation layer, and $\mathbf{h}_i$ is the layer weights. Now the dimension of r_i is the same as embedding dimension m in $R(t)$.

We used r_i to calculate attention weights using what is called *general attention* [21]:

$$\mathbf{a}_i = \text{softmax}(r_i^\top \mathbf{W} R(t)_j) \tag{8}$$

where $\mathbf{W}$ is a learnable weighting matrix, $R(t)_j$ is the j-th column of the social data matrix, and $\mathbf{a}_i = \{a_{i1}, ..., a_{im}\}$.

Then we calculate the output of the social context encoder module, a context vector c_i for item i

$$\mathbf{c}_i = \sum_j a_{ij} R(t)_j. \tag{9}$$

In this way, the context vector c_i contains both the social data information and the item information. Finally, we use linear regression to convert all contained information into the social latent factor:

$$s_{i,t} = w^\top c_i + b, \tag{10}$$

where w and b are learnable parameters.

4 Experimental Evaluation

We conduct experimental evaluations on two pairs of real-world datasets, in order to test the effect on recommendation accuracy and bias reduction, when applying our method and compared methods. In this section, we first introduce the datasets used and the experimental setup. Then we discuss the evaluation results.

4.1 Datasets

We test our approach intently on e-commerce items that are not famous or are not linkable to famous entities. Therefore we prepare two such e-commerce datasets. The first dataset is provided by our industry partner, which is a flash sales (FS) dataset generated in a Japanese e-commerce platform. It includes daily-life items such as food, cosmetics, and home appliances. The second dataset is a subset of the well-known Amazon dataset [23], containing only the category of grocery, which includes food, drinks, and other daily-life items. Both datasets include a number of users, items, and timestamped purchase records. The FS dataset is of a four-month period between June and September 2017. The Amazon dataset is of a six-month period between January and June 2018.

In the same period as the e-commerce datasets, we prepare corresponding social data datasets. For the FS dataset, we collect tweets from a number of Japanese Twitter users through the timeline API[2]. For the Amazon dataset, we use the publicly available Reddit dataset[3] of the same period. The number of users, items, interactions, and tweets for each pair of the datasets are shown in Table 1.

We divide the datasets into training and test datasets. For the FS+Twitter dataset, we use the first three months as the training data and the last month as the test data. For the Amazon+Reddit dataset, we use the first four months as the training data, and the last two months as the test data.

[2] https://developer.twitter.com/en/docs/twitter-api/tweets/timelines/introduction.
[3] https://files.pushshift.io/reddit/submissions/.

Table 1. Dataset sizes of e-commerce and social data

	User	Item	Interaction	Tweets
FS+Twitter	40,895	20,542	125,968	2.4M
Amazon+Reddit	18,659	11,291	57,489	69M

4.2 Experimental Setup

We test our model with several other recommendation models, including a state-of-the-art debiased model and a social data-based model.

- MF [14]. The well-known matrix factorization model that produces two lower-rank matrices that represent latent user factor and item factor. It lacks user and item biases compared to SVD.
- Popularity-bias Deconfounding (PopDecon) [30]. This is a state-of-the-art debiased recommendation model. We implement it with BPR-MF according to the description in the paper.
- SVD. The base recommendation model.
- TimeSVD [19]. This is a time-sensitive SVD model. We implement the period-based version of the model since there is no time overlapping between training and test datasets in our evaluation.
- Item Social Trend Encoder (ISTE) [32]. This is a state-of-the-art recommendation model that integrates real-time social data. It differs from our model in that it uses NeuMF [13] architecture and concatenates social data as a vector in the output layer.

We implement our model and all compared models as neural networks using Python and Tensorflow[4]. We use 20 as the embeddings size for all MF and SVD models.

We follow the established method and measure recommendation accuracy as Recall and Normalized Discounted Cumulative Gain (NDCG) [16]. Both measure the ranking quality when given a positive test case and a number of negative candidates. Recall is based on hit or not-hit, while NDCG is based on the position of the positive item in the ranking.

For measuring bias reduction, we follow an existing approach [30]. We first divide the items into 10 groups according to the order of their interaction number in the training data. Group 1 contains the most popular items, group 10 contains the least popular items, and so on. At the same time, we keep the total number of training interactions in each group about the same, which means group 1 has fewer items and more interactions per item, etc. If a recommendation model is not biased, the recommended items should have the same distribution over groups as in the training data, which is 0.1 for all groups. Deviation from this distribution indicates recommendation bias. A lower deviation generally means lower recommendation bias. Like [30], we measure the deviation as the standard deviation.

[4] The source code will be made available soon.

4.3 Results Analysis

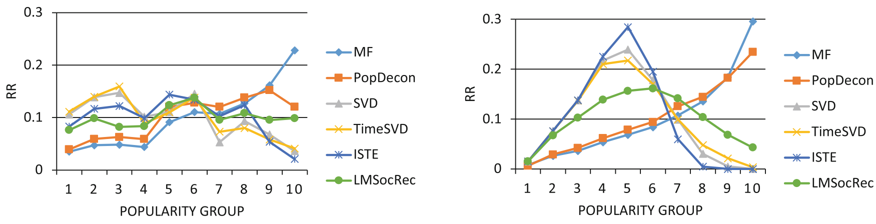
First we look at the impact of different models on recommendation accuracy, which is shown in Table 2. We can see that SVD models are generally better than MF models (MF and PopDecon) after considering user and item biases. Our method of integrating social data improves NDCG@5 by 25% and 10% for two datasets, compared to the original SVD. TimeSVD gives time context to the item like our method but not the social context, and it turns out this is not helpful for improving recommendation accuracy. ISTE outperforms other baselines by considering social context, which other baselines do not have. Our method, though, is better than ISTE, even though the latter uses a more complex architecture. So our model is successful in terms of improving recommendation accuracy.

Table 2. Recommendation accuracy of different models

	FS+Twitter		Amazon+Reddit	
	Recall@5	NDCG@5	Recall@5	NDCG@5
MF	0.154	0.108	0.112	0.075
PopDecon	0.237	0.165	0.145	0.104
SVD	0.335	0.243	0.269	0.178
TimeSVD	0.315	0.228	0.263	0.174
ISTE	0.334	0.245	0.269	0.173
LMSocRec	**0.342**	**0.265**	**0.351**	**0.256**

Next, we look at the impact on bias reduction. Figure 3 shows the distribution of item recommendations over popularity groups. We can see for both datasets, the methods operate in two modes. The MF-based methods tend to recommend the least popular items, because these are the majority of the items in the test candidates. The SVD-based methods tend to recommend items with popularity somewhere in the middle, especially popularity group 5. That is because these methods learned popular item bias based on the training data. Both types of methods do not recommend the most popular items in the training data because the market is changing for both datasets, and popular items in the training period do not appear in the test period. Within the SVD-based methods, we can see the proposed LMSocRec furthermore recommends more diversified items, and the distribution over popularity groups is closer to the training distribution, which is constant at 0.1 over groups.

We can also look at the single value summary, i.e., the deviation score. Table 3 shows the standard deviation of the recommendation distribution for different models. We can see PopDecon method, specialized in debiasing, achieves lower deviation, especially for the Amazon dataset. However, the lowest deviation achieved is by our proposed method, indicating its effectiveness in bias reduction and recommendation diversification.



(a) Flash Sales Dataset (b) Amazon Dataset

Fig. 3. Recommendation rate over popularity groups

Table 3. Standard deviation of recommendation distribution over popularity groups

	MF	PopDecon	SVD	TimeSVD	ISTE	LMSocRec
FS	0.062	0.040	0.038	0.038	0.038	**0.019**
Amazon	0.087	0.072	0.090	0.080	0.105	**0.050**

5 Conclusion

Social data that represents the whole social environment has been aiding many ubiquitous tasks. In this paper, we attempt to use social data to debias recommendation models. One major cause for a recommendation system to generate biased recommendations is that the system is isolated in a platform, and the machine learning model is trained on a feedback loop. In contrast to existing debiasing studies, we propose a method to bring out-of-system social data into the recommendation system to improve recommendation performance. The method integrates social data, represented as embeddings generated by a pre-trained language model, as a latent factor into the traditional SVD model. Experimental evaluations with two pairs of e-commerce plus social data datasets show that our method is successful, outperforming state-of-the-art debiasing methods in terms of both recommendation accuracy and bias reduction. In the future, we will continue to investigate more effective ways of social data integration.

References

1. Abdollahpouri, H.: Popularity bias in ranking and recommendation. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pp. 529–530 (2019)
2. Amin, S., Uddin, M.I., AlSaeed, D.H., Khan, A., Adnan, M.: Early detection of seasonal outbreaks from twitter data using machine learning approaches. Complexity **2021** (2021)
3. Birhane, A., Kasirzadeh, A., Leslie, D., Wachter, S.: Science in the age of large language models. Nat. Rev. Phys. 1–4 (2023)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**(Jan), 993–1022 (2003)

5. Brown, T., et al.: Language models are few-shot learners. Adv. Neural Inf. Process. Syst. **33**, 1877–1901 (2020)
6. Chaney, A.J., Stewart, B.M., Engelhardt, B.E.: How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 224–232 (2018)
7. Chen, C., et al.: An efficient adaptive transfer neural network for social-aware recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 225–234 (2019)
8. Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., He, X.: Bias and debias in recommender system: a survey and future directions. ACM Trans. Inf. Syst. (2020)
9. Davidson, J., et al.: The youtube video recommendation system. In: Proceedings of the fourth ACM Conference on Recommender Systems, pp. 293–296 (2010)
10. Deng, Z., Yan, M., Sang, J., Xu, C.: Twitter is faster: personalized time-aware video recommendation from twitter to youtube. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **11**(2), 1–23 (2015)
11. Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., Cohen, W.W.: Tweet2Vec: character-based distributed representations for social media. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers (2016)
12. Floridi, L., Chiriatti, M.: GPT-3: its nature, scope, limits, and consequences. Mind. Mach. **30**, 681–694 (2020)
13. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182, Perth, Australia, 2017. ACM (2017)
14. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 549–558 (2016)
15. Jannach, D., Lerche, L., Kamehkhosh, I., Jugovac, M.: What recommenders recommend: an analysis of recommendation biases and possible countermeasures. User Model. User-Adap. Inter. **25**(5), 427–491 (2015)
16. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. (TOIS) **20**(4), 422–446 (2002)
17. Kalampokis, E., Tambouris, E., Tarabanis, K.: Understanding the predictive power of social media. Internet Res. (2013)
18. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)
19. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 447–456 (2009)
20. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880 (2020)
21. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421 (2015)
22. Misztal-Radecka, J., Indurkhya, B.: A bias detection tree approach for detecting disparities in a recommendation model's errors. User Model. User-Adapt. Interact. 1–37 (2022)

23. Ni, J., Li, J., McAuley, J.: 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), pp. 188–197 (2019)
24. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th International World Wide Web Conference, pp. 851–860 (2010)
25. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
26. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 950–958 (2019)
27. Wu, S., Hofman, J.M., Mason, W.A., Watts, D.J.: Who says what to whom on twitter. In: Proceedings of the 20th International World Wide Web Conference, pp. 705–714 (2011)
28. Yamada, K.: Detecting two types of seasonal words using simple autocorrelation analysis. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 3162–3167. IEEE (2017)
29. Yates, A., Nogueira, R., Lin, J.: Pretrained transformers for text ranking: bert and beyond. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 1154–1156 (2021)
30. Zhang, Y., et al.: Causal intervention for leveraging popularity bias in recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 11–20 (2021)
31. Zhang, Y., Hara, T.: Explainable integration of social media background in a dynamic neural recommender. ACM Trans. Knowl. Discov. Data (TKDD) **17**(3), 1–14 (2023)
32. Zhang, Y., Maekawa, T., Hara, T.: Using social media background to improve cold-start recommendation deep models. In: Proceedings of 2021 IEEE International Joint Conference on Neural Networks IJCNN, pp. 1–8 (2021)
33. Zhao, W.X., Li, S., He, Y., Chang, E.Y., Wen, J.-R., Li, X.: Connecting social media to e-commerce: cold-start product recommendation using microblogging information. IEEE Trans. Knowl. Data Eng. **28**(5), 1147–1159 (2015)