Combining Unstructured Content and Knowledge Graphs into Recommendation Datasets

Weizhe Lin[1,†], Linjun Shou[2], Ming Gong[2], Pei Jian[3], Zhilin Wang[4], Bill Byrne[1] and Daxin Jiang[2]

[1]Department of Engineering, University of Cambridge, Cambridge, United Kingdom
[2]Microsoft STCA, Beijing, China
[3]Simon Fraser University, British Columbia, Canada
[4]University of Washington, Seattle, United States

Abstract

Popular book and movie recommendation datasets can be associated with Knowledge Graphs (KG) that enable the development of KG-based recommender systems. However, most of these approaches are based on Collaborative Filtering, leaving Content-based Filtering approaches unexploited. This is partially due to the lack of items' content-based information (e.g. summary texts of movies and books) in datasets. To facilitate the research in achieving both KG-aware and content-aware recommender systems, we contribute to public domain resources through the creation of a large-scale Movie-KG dataset and an extension of the already public Amazon-Book dataset through incorporation of text descriptions crawled from external sources. Both datasets provide items' descriptive texts that enable recommendations based on unstructured content. We provide benchmark results as well as showing the value of the content-based information in making recommendations.

Keywords

Knowledge graph, recommender systems, recommendation dataset

1. Introduction

In recent years, modern recommender systems (RS) based on deep learning models are considered as the most successful solutions that recommend items (such as movies, books, and news) to users. There has been a growing interest in incorporating Knowledge Graphs (KG) into recommendation since recent study showed that RS can benefit from the external information provided by KGs to enrich the user/item representations [1]. KGs link items to be recommended to other related KG entities, and these connections serve as "item properties" (e.g. movie genre, movie actors, and book editors). Datasets such as Amazon-Book and MovieLens-20M provide user-item interactions along with a KG that provides external knowledge. This makes reasoning and searching over KGs possible.

While KGs can readily incorporate structured content information and external knowledge, *unstructured* content such as item descriptions, is unexploited in these popular KG-based recommendation datasets. We note that recent Transformer-based models, such as BERT [2] and GPT-2 [3], have been widely used in news recommendation for extracting descriptive content from natural language, and they have been shown to be effective. We take the view that summary texts of books and movies can also help improve KG-aware systems that solely rely on interactions and KGs. For example, the two movies "Interstellar" and "Inception", have a very similar set of structured properties including genre, writer, and director, but their descriptions provide more fine-grained discriminative information, making it clear that one is about physics and universe and the other is about adventures and dreams. Content-aware systems can be built from such information.

Therefore, in this work, we contribute to the research community through the creation of two new datasets for movie and book recommendations: (1) an extension of the widely-used Amazon-Book dataset through the addition of summary texts for books and (2) a new large-scale movie recommendation dataset. They both consist of (1) large-scale user-item interactions; (2) up-to-date, large-scale enormous knowledge graphs; and (3) rich descriptive attributes suitable for content-based movie and book recommendation. Based on these two realistic datasets, we provide benchmarking results with selected strong *Collaborative Filtering* (CF) and *Content-based Filtering* (CBF) models.

In summary, our contributions are:

1. We extend the popular book recommendation dataset Amazon-Book by extracting book sum-

4th Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) Workshop @ RecSys 2022, September 18–23 2023, Seattle, WA, USA.

[†]This work was done during Weizhe Lin's internship at Microsoft STCA.

wl356@cam.ac.uk (W. Lin); lisho@microsoft.com (L. Shou); migon@microsoft.com (M. Gong); jpei@cs.sfu.ca (P. Jian); zhilinw@uw.edu (Z. Wang); bill.byrne@eng.cam.ac.uk (B. Byrne); djiang@microsoft.com (D. Jiang)

mary texts from web resources.

2. We have created a brand-new large-scale dataset for movie recommendation. The dataset was collected from real users of Microsoft Edge, and KGs and movie descriptions were extracted from internal knowledge bases.
3. We finally provide benchmark results for both datasets with strong RS models. The results show that the incorporation of unstructured content leads to system improvements.

2. Related Work

Recommendation Datasets. Some popular existing recommendation datasets are shown in the first section of Table 1. **MovieLens-20M**[1] is a popular benchmark that has been widely used, while **Flixster**[2] is large but less popular in recent work. They are both movie datasets but they do not provide associated KGs to enable the development of KG-aware systems. **Amazon-Book** [4], **Last-FM** [4], **Book-Crossing** [5][3] and **Alibaba-iFashion** [6] are dedicated for KG-based CF model evaluation. They provide rich user-item interactions but unstructured content-based information (e.g. item summary texts in natural language) is not included. In news recommendations, in contrast to the small Yahoo! dataset [7], MIND [8] from Microsoft is a more recent and larger dataset collected from real user behaviors. However, these datasets only support the development of content-aware CBF models since no KG is available.

KG-aware Recommender Systems. Beyond traditional CF models that model users and items through only interaction data [9, 10, 11, 12, 13, 14, 15, 13, 16, 17, 18, 19, 20, 21], KG-based CF models fuse external knowledge from auxiliary KGs to improve both the accuracy and explainability of recommendation [1, 22]. The CF methods to exploit the KGs can be categorized into *Embedding-based Methods* [23, 24, 25], *Path-based Methods* [26, 27, 28, 29], and *GNN-based Methods* [30, 31, 27, 4, 32]. CBF models match items to a user by considering the metadata (content-based information) of items with which the user has interacted [33, 34, 35, 36, 37], while most research in KG-based CBF, a recently popular topic, focuses on enhancing the item representations with KG embeddings by mapping relevant KG entities to the content of items, e.g., by entity linking [38, 39].

3. Datasets

We introduce the collection process of the two new datasets in this section: (1) a large-scale high-quality movie recommendation dataset that was newly collected from real user behaviors (Sec. 3.1); (2) the Amazon-Book-Extended dataset based on the Amazon-Book dataset [4] and newly augmented with textual book descriptions (Sec. 3.2).

3.1. Movie KG Dataset

3.1.1. Dataset Construction

This dataset was formed by real user browsing behaviors that were logged by a popular commercial browser between 13/05/2021 and 20/06/2021 (39 days in total). Sensitivity of data and privacy was firstly removed by decoupling users' real identity (e.g. IP, account identifier) from the data and assigning each user an insensitive unique virtual identifier. To link users' browsing history with movies, we exploited a commercial knowledge graph that consists of a large number of entities intended to cover the domain and a rich set of relationships among them. Entities with type "movie" were extracted together with their one-hop neighbors and edges. In this way a movie-specific sub-graph was extracted from the original knowledge graph.

Drawing from billions of user browsing logs, KG entity linking was performed to match web page titles with the movie titles in the movie-specific KG. Movie entities were extracted using an internal service that provides Named Entity Recognition and entity linking. To make the dataset more compact: (1) consecutive and duplicated user interactions with the same movie were merged to one single record by aggregating their browsing times; (2) less active users with fewer than 10 interactions in 39 days were dropped; (3) the most frequent 50,000 movies with the most user interactions were selected, since tail records were considered unreliable.

125,218 active users and 50,000 popular movies were chosen. KG entities related to these movies were extracted to form a sub-KG with 250,327 entities (including movie items). Each user is represented by [$UserID$, $UserHistory$], where $UserID$ is a unique identifier that has been delinked from real user identity, and $UserHistory$ is an ID list of movie items that the user has browsed.

For training and evaluating recommender systems, $UserHistory$ was split into train/validation/test sets: the first 80% of users' historical interactions (ordered by click date-time) are in the train set; 80% to 90% serves as the validation set; and the remaining is reserved for testing. During validation and test, interactions in the train set serve as users' previous clicked items, and systems are evaluated on their ratings for test set items. In addition to the traditional data split strategy, a **cold-start user set** was created to evaluate model performance on users outside the training dataset. 3% of users were moved to the cold-start set and they are not available in model

[1]https://grouplens.org/datasets/movielens/
[2]https://sites.google.com/view/mohsenjamali/flixter-data-set
[3]http://www2.informatik.uni-freiburg.de/ cziegler/BX/

Table 1
Comparison with existing popular datasets for recommender systems. Ent:Entity; Rel:Relation; UnCont: Unstructured Content.

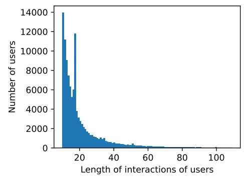
Type	Dataset	#Users	#Items	#Interactions	KG	#Ent.	#Rel.	#Triplets	UnCont.
Movie	MovieLens-20M	138,000	27,000	20,000,000	No	N/A	N/A	N/A	No
Moive	Flixster	1,002,796	66,730	1,048,576	No	N/A	N/A	N/A	No
Book	Amazon-Book	70,679	24,915	847,733	Yes	88,572	39	2,557,746	No
Book	Book-Crossing	276,271	271,379	1,048,575	Yes	25,787	18	60,787	No
Music	Last-FM	23,566	48,123	3,034,796	Yes	58,266	9	464,567	No
Shopping	Alibaba-iFashion	114,737	30,040	1,781,093	Yes	59,156	51	279,155	No
News	Yahoo!	-	14,180	34,022	No	N/A	N/A	N/A	Yes
News	MIND	1,000,000	161,013	24,155,470	No	N/A	N/A	N/A	Yes
Book	Amazon-Book-Extended (Ours)	70,679	24,915	847,733	Yes	88,572	39	2,557,746	Yes
Movie	Movie-KG-Dataset (Ours)	125,218	50,000	4,095,146	Yes	250,327	12	12,055,581	Yes

training. To challenge systems' real abilities under extreme cold-start scenarios, we chose users that have very few interactions (lower than 20 interactions per user on average). This portion of interactions was split into a cold-start history set with the first 80% interactions of each user, and a test set with the remaining 20%.

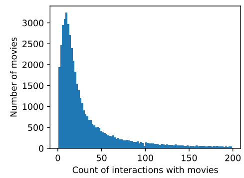
A comparison with some popular existing recommendation datasets is provided in Table 1. Our new movie dataset is based on large user populations and movie inventory and draws from an extensive movie-specific KG with millions of triplets. This is a rich resource not yet provided by existing popular movie recommendation datasets. Our dataset provides not only titles and genres (as in MovieLens-20M), but also other content-based information, such as rich summary texts of movies that enable content-based recommendation with large language understanding models such as BERT and GPT-2.

3.1.2. Statistical Analysis

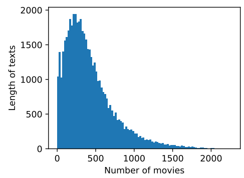
Fig. 1 presents the key statistics of the dataset. This dataset contains 125,218 users, 50,000 movie items, and 4M+ interactions. The KG contains 250,327 entities with 12 relation types and 12,055,581 triplets. The attributes of movies include content-based properties (title, description, movie length) and properties already tied to knowledge graph entities (production company, country, language, producer, director, genre, rating, editor, writer, honors, actor). Fig. 1(a)-1(c) shows the distribution of length of interactions with movies per user, number of interactions with users per movie, and the length of movie descriptions. The average length of movie descriptions is 425 words, which is long enough for models to encode movies from the summary of stories. The average number of user interactions is 32.70, providing rich resources for modeling long-term user interest. The first 19 days of data (with similar data distribution but fewer users and interactions) is separated as "Standard" for most research purposes, while the full 39 days of data are released as "Extended" for more extended use, such as training industrial products [4].



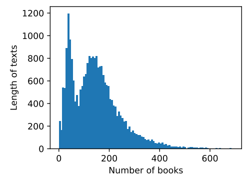
(a) Length of user history ($\mu = 32.70$, $\sigma = 45.76$)



(b) Users' interactions with each movie ($\mu = 83.33$, $\sigma = 582.50$)



(c) Length of movie description in Movie-KG-Dataset ($\mu = 425.22$, $\sigma = 329.93$)



(d) Length of book description in Amazon-Book-Extended ($\mu = 151.75$, $\sigma = 206.98$)

Figure 1: Key statistics of Movie-KG-Dataset (Fig. 1(a)-1(c)) and Amazon-Book-Extended (Fig. 1(d)).

3.2. Extended Amazon Book Dataset

The Amazon-Book dataset was originally released by [4] and has been used in the development of many advanced recommendation systems. However, this dataset provides only CF interaction data without content-based features needed for CBF evaluation. To fill this gap, we extend the dataset with descriptions extracted from multiple data sources, noting that the dataset was originally collected in 2014 and many items have since expired on the Amazon website. The procedure is summarized as follows:

(1) We matched product descriptions in Amazon with entries in Amazon-Book using their unique item iden-

[4]We provide results of this extended set in our repository.

Table 2
Benchmark results on the Amazon-Book-Extended dataset. Best performance of the proposed models is marked in **bold**. The average of 3 runs is reported to mitigate experimental randomness.

	Recall			ndcg			Hit Ratio		
	@20	@60	@100	@20	@60	@100	@20	@60	@100
BPRMF	0.1352	0.2433	0.3088	0.0696	0.09568	0.1089	0.2376	0.3984	0.4816
CKE	0.1347	0.2413	0.3070	0.0691	0.09482	0.1081	0.2373	0.3963	0.4800
KGAT	0.1527	0.2595	0.3227	0.0807	0.10655	0.1194	0.2602	0.4156	0.4931
KGIN	0.1654	0.2691	0.3298	0.0893	0.1145	0.1267	0.2805	0.4289	0.5040
KMPN	0.1719	0.2793	0.3405	0.0931	0.1189	0.1315	0.2910	0.4421	0.5166
NRMS-BERT	0.1142	0.2083	0.2671	0.0592	0.0817	0.0935	0.2057	0.3487	0.4273
Mixture of Expert (KMPN + NRMS-BERT)	**0.1723**	0.2791	0.3281	**0.0933**	0.1161	0.1214	**0.2913**	0.4425	0.5022
CKMPN	0.1718	**0.2821**	**0.3460**	0.0928	**0.1197**	**0.1326**	0.2908	**0.4474**	**0.5244**

tifiers (asin). 24,555 items were successfully matched (98.56%); (2) we then matched each of the remaining items with the most relevant entry in a huge commercial KG which enables rich descriptions to be extracted. 325 items were matched at this step; (3) the remaining 28 items were matched manually to the most relevant products in Amazon.

The distribution of description lengths is given in Fig. 1(d).

4. Benchmarking

4.1. Evaluation Metrics

Following common practice [19, 4, 6, 40], we report metrics for evaluating model performance: (1) *Recall@K*: within top-K recommendations, how well the system recalls the test-set browsed items for each user; (2) *ndcg@K* (Normalized Discounted Cumulative Gain) [40]: increases when relevant items appear earlier in the recommended list; (3) *HitRatio@K*: how likely a user finds at least one interesting item in the recommended top-K items.

4.2. Baselines

We take the performance of several recently published recommender systems as points for comparison[5]. We carefully reproduced all these baseline systems from their repositories[6].

BPRMF [10]: a powerful Matrix Factorization (MF) method that applies a generic optimization criterion BPR-Opt for personalized ranking. Limited by space, other MF models (e.g. FM [41], NFM [13]) are not presented since BPRMF outperformed them.

CKE [24]: a CF model that leverages heterogeneous information in a knowledge base for recommendation.

KGAT [4]: Knowledge Graph Attention Network (KGAT) which explicitly models high-order KG connectivities in KG. The models' user/item embeddings were initialized from the pre-trained **BPRMF** weights.

KGIN [6]: a state-of-the-art KG-based CF model that models users' latent intents (preferences) as a combination of KG relations.

KMPN [42]: a KG-based CF model that models users through learning a set of preference embeddings.

NRMS-BERT [42]: a strong CBF model that incorporates a pre-trained BERT for extracting content-based features from natural language. It was inspired by NRMS [34], a strong news recommender system with a bi-encoder architecture.

Mixture of Expert: a hybrid system where the output scores of two systems, KMPN and NRMS-BERT, passed through 3 layers of a Multi-Layer Perception (MLP) to obtain final item ratings.

CKMPN [42]: a contrastive learning approach that fuses the features of NRMS-BERT with those of KMPN in training. It is also a hybrid system that leverages both CBF and CF features.

4.3. Discussion

On the Amazon-Book-Extended dataset, as shown in Table 2, KG-based systems (e.g. KGAT, KMPN) leverage structured information embeded in KGs to achieve ~0.17 Recall@20. The CBF-system (NRMS-BERT) achieves 0.1142 Recall@20 with only summary texts of books. The performance is not far from that of KG-based models. It shows that our extension to the original dataset is successful and this content-based information can be used for making content-aware recommendations. The best scores are obtained by hybrid methods (MoE and CKMPN) that combine KMPN and NRMS-BERT to make recommendations. In particular, the CKMPN model improves results of @60/@100, showing that even though NRMS-BERT's performance is lower, KG-based systems can still benefit from incorporating its features. This

[5]They are also baseline systems being compared in a recent paper [6] (WWW'21).

[6]As a result, the results reported here may differ from those of the original papers.

Table 3
Benchmark results on Movie-KG-Dataset. Best performance of the proposed models is marked in **bold**. The average of 3 runs is reported to mitigate experimental randomness.

	Recall			ndcg			Hit Ratio		
	@20	@60	@100	@20	@60	@100	@20	@60	@100
BPRMF	0.1387	0.1944	0.2206	0.0961	0.1137	0.1192	0.1980	0.2785	0.3236
CKE	0.1369	0.1898	0.2150	0.0940	0.1108	0.1160	0.1950	0.2707	0.3155
KGAT	0.1403	0.1928	0.2185	0.1006	0.1173	0.1226	0.1997	0.2742	0.3196
KGIN	0.1351	0.2119	0.2445	0.0982	0.1254	0.1322	0.2194	0.3081	0.3643
KMPN	0.1434	0.2130	0.2427	0.1073	0.1305	0.1367	0.2193	0.3098	0.3602
NRMS-BERT	0.1241	0.1669	0.1890	0.1034	0.1213	0.1257	0.1728	0.2369	0.2773
CKMPN	**0.1457**	**0.2157**	**0.2462**	**0.1149**	**0.1417**	**0.1482**	**0.2266**	**0.3153**	**0.3668**
On the cold-start test set									
KMPN	0.1019	0.1672	0.2122	0.0561	0.0710	0.0806	0.1783	0.2812	0.3414
NRMS-BERT	0.0437	0.0911	0.1253	0.0206	0.0314	0.0381	0.0807	0.1580	0.2071
CKMPN	0.1024	0.1741	0.2130	0.0570	0.0729	0.0808	0.1812	0.2839	0.3380

also shows that content-based features are useful, but simple aggregation schemes can not improve the performance significantly. One research challenge left for future research is how to better exploit the content-based features.

We observe a similar trend on Movie-KG-Dataset. The hybrid method, CKMPN, achieves the best performance. We note that the performance of NRMS-BERT is closer to that of KG-based models (e.g. KMPN). The improvement brought by content-based features is also more obvious on Movie-KG-Dataset. This is because Movie-KG-Dataset has larger average text lengths compared to Amazon-Book-Extended (425.22 v.s. 151.75), offering richer information that result in more discriminative item embeddings.

An example output of systems is presented in Table 4. Y/N indicates whether or not the movie appears in the top-100 recommendation list of the four models (KMPN/NRMS-BERT/Mixture of Expert (MoE)/CKMPN). This user has browsed Tenet (2020) directed by Christopher Nolan. The movie Source Code (2011) and Tenet are both about time travel, but they have quite different film crews. As a result, Source Code was considered positive by NRMS-BERT which evaluates on the movie description, but was considered negative by KG-based KMPN. Combining the scores of both systems, MoE did not recommend the movie. However, CKMPN complemented the failure of KMPN and gave a high score for this movie, by learning a content-aware item representation based on the representation of NRMS-BERT through contrastive learning. In contrast, Dunkirk (2017) is about war and history which is not in the same topic as Tenet. However, since they were directed by the same director, KMPN and CKMPN both recommended this movie, while MoE's prediction was negatively affected by NRMS-BERT. This case study suggests that both KGs and unstructured content (summary texts of movies in this case) are useful for making recommendations. It is challenging to exploit them at the same time while overcoming the limitations of each item property.

Table 4
Case study for a user who have browsed the movie Tenet (2020). Source Code (2011) has a similar genre, while Dunkirk (2017) has the same director. Y/N: whether or not the movie appears in the top-100 recommendation list of the models. MoE: Mixture of Expert.

Item	KMPN	NRMS-BERT	MoE	CKMPN
Source Code (2011)	N	Y	N	Y
Dunkirk (2017)	Y	N	N	Y

On the cold-start test set, CF, CBF, and hybrid methods have achieved much lower results. The performance reduction is more obvious on the CBF model (NRMS-BERT drops from 0.1241 to 0.0437 Recall@20), since the users in this set have very few (<20) historical interactions on average, making a content-based system fail to capture their characterstics and make suitable recommendations. Therefore, another research challenge is to improve the cold-start performance of all these models, making models more robust to extreme cases.

5. Conclusion

To facilitate research in developing RS models that are both KG-aware and content-aware, we introduced two datasets: (1) Amazon-Book-Extended which inherits a popular book recommendation dataset and has been newly extended with summary descriptions; (2) a new large-scale movie recommendation dataset, Movie-KG-Dataset, based on recently collected user interactions on a widely-used commercial web browser, accompanied by a knowledge graph and summary texts for the movies.

We provided benchmark results that demonstrated the value of incorporating content-based item descriptions. Both datasets are shared in our github repository: https://github.com/LinWeizheDragon/Content-Aware-Knowledge-Enhanced-Meta-Preference-Networks-for-Recommendation.

References

[1] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, IEEE Transactions on Knowledge and Data Engineering (2020).

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[4] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 950–958.

[5] C.-N. Ziegler, S. M. McNee, J. A. Konstan, G. Lausen, Improving recommendation lists through topic diversification, in: Proceedings of the 14th international conference on World Wide Web, 2005, pp. 22–32.

[6] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, T.-S. Chua, Learning intents behind interactions with knowledge graph for recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 878–887.

[7] Z. Yang, C. Xu, W. Wu, Z. Li, Read, attend and comment: A deep architecture for automatic news comment generation, in: 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019.

[8] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu, et al., Mind: A large-scale dataset for news recommendation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3597–3606.

[9] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09, AUAI Press, Arlington, Virginia, USA, 2009, p. 452–461.

[11] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 426–434. URL: https://doi.org/10.1145/1401890.1401944. doi:10.1145/1401890.1401944.

[12] S. Rendle, Factorization machines with libfm, ACM Transactions on Intelligent Systems and Technology (TIST) 3 (2012) 1–22.

[13] X. He, T.-S. Chua, Neural factorization machines for sparse predictive analytics, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 355–364. URL: https://doi.org/10.1145/3077136.3080777. doi:10.1145/3077136.3080777.

[14] R. J. Oentaryo, E.-P. Lim, J.-W. Low, D. Lo, M. Finegold, Predicting response in mobile advertising with hierarchical importance-aware factorization machine, in: Proceedings of the 7th ACM international conference on Web search and data mining, 2014, pp. 123–132.

[15] H. Guo, R. TANG, Y. Ye, Z. Li, X. He, Deepfm: A factorization-machine based neural network for ctr prediction, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 1725–1731. URL: https://doi.org/10.24963/ijcai.2017/239. doi:10.24963/ijcai.2017/239.

[16] W. Zhang, T. Du, J. Wang, Deep learning over multi-field categorical data, in: European conference on information retrieval, Springer, 2016, pp. 45–57.

[17] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah, Wide deep learning for recommender systems, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS 2016, Association for Computing Machinery, New York, NY, USA, 2016, p. 7–10. URL: https://doi.org/10.1145/2988450.2988454. doi:10.1145/2988450.2988454.

[18] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, J. Wang, Product-based neural networks for user

response prediction, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 1149–1154.

[19] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 173–182.

[20] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 974–983.

[21] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.

[22] J. Chicaiza, P. Valdiviezo-Diaz, A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions, Information 12 (2021) 232.

[23] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: The World Wide Web Conference, 2019, pp. 2000–2010.

[24] F. Zhang, N. J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 353–362.

[25] Y. Cao, X. Wang, X. He, Z. Hu, T.-S. Chua, Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences, in: The world wide web conference, 2019, pp. 151–161.

[26] B. Hu, C. Shi, W. X. Zhao, P. S. Yu, Leveraging meta-path based context for top-n recommendation with a neural co-attention model, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1531–1540.

[27] J. Jin, J. Qin, Y. Fang, K. Du, W. Zhang, Y. Yu, Z. Zhang, A. J. Smola, An efficient neighborhood-based interaction model for recommendation on heterogeneous graph, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 75–84.

[28] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, J. Han, Personalized entity recommendation: A heterogeneous information network approach, in: Proceedings of the 7th ACM international conference on Web search and data mining, 2014, pp. 283–292.

[29] H. Zhao, Q. Yao, J. Li, Y. Song, D. L. Lee, Meta-graph based recommendation fusion over heterogeneous information networks, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 635–644.

[30] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, Z. Wang, Knowledge-aware graph neural networks with label smoothness regularization for recommender systems, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 968–977.

[31] H. Wang, M. Zhao, X. Xie, W. Li, M. Guo, Knowledge graph convolutional networks for recommender systems, in: The World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 3307–3313. URL: https://doi.org/10.1145/3308558.3313417. doi:10.1145/3308558.3313417.

[32] Z. Wang, G. Lin, H. Tan, Q. Chen, X. Liu, Ckan: Collaborative knowledge-aware attentive network for recommender systems, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 219–228.

[33] J. Liu, P. Dolan, E. R. Pedersen, Personalized news recommendation based on click behavior, in: Proceedings of the 15th international conference on Intelligent user interfaces, 2010, pp. 31–40.

[34] C. Wu, F. Wu, S. Ge, T. Qi, Y. Huang, X. Xie, Neural news recommendation with multi-head self-attention, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 6389–6394. URL: https://aclanthology.org/D19-1671. doi:10.18653/v1/D19-1671.

[35] S. Okura, Y. Tagami, S. Ono, A. Tajima, Embedding-based news recommendation for millions of users, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 1933–1942.

[36] J. Lian, F. Zhang, X. Xie, G. Sun, Towards better representation learning for personalized news recommendation: a multi-channel deep fusion approach., in: IJCAI, 2018, pp. 3805–3811.

[37] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, X. Xie, Npa: neural news recommendation with personalized attention, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2576–2584.

[38] D. Liu, J. Lian, S. Wang, Y. Qiao, J.-H. Chen, G. Sun, X. Xie, Kred: Knowledge-aware document representation for news recommendations, in: Fourteenth

ACM Conference on Recommender Systems, 2020, pp. 200–209.
[39] H. Wang, F. Zhang, X. Xie, M. Guo, Dkn: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 world wide web conference, 2018, pp. 1835–1844.
[40] W. Krichene, S. Rendle, On sampled metrics for item recommendation, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1748–1757.
[41] S. Rendle, Z. Gantner, C. Freudenthaler, L. Schmidt-Thieme, Fast context-aware recommendations with factorization machines, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, Association for Computing Machinery, New York, NY, USA, 2011, p. 635–644. URL: https://doi.org/10.1145/2009916.2010002. doi:10.1145/2009916.2010002.
[42] W. Lin, L. Shou, M. Gong, P. Jian, Z. Wang, B. Byrne, D. Jiang, Transformer-empowered content-aware collaborative filtering, in: 4th Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) Workshop @ RecSys 2022, 2022.
[43] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6980.

A. Training Details

All experiments were run on 8 NVIDIA A100 GPUs with batch size 8192×8 for KMPN/CKMPN and 4×8 for NRMS-BERT. Adam [43] is used to optimize models. KMPN/CKMPN is trained for 2000 epochs with linearly decayed learning rates from 10^{-3} to 0 for Amazon-Book-Extended and 5×10^{-4} to 0 for Movie-KG-Dataset. NRMS-BERT is trained for 10 epochs at a constant learning rate of 10^{-4}.

We follow the official repository of KGAT[7] for BPRMF, CKE, and KGAT training.

Codes and pre-trained models are released in https://github.com/LinWeizheDragon/Content-Aware-Knowledge-Enhanced-Meta-Preference-Networks-for-Recommendation.

[7]https://github.com/xiangwang1223/knowledge_graph_attention_network