Transformer-Empowered Content-Aware Collaborative Filtering

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

Knowledge graph (KG) based Collaborative Filtering (CF) is an effective approach to personalize recommender systems for relatively static domains such as movies and books, by leveraging structured information from KG to enrich both item and user representations. This paper investigates the complementary power of unstructured content information (e.g. rich summary texts of items) in KG-based CF recommender systems. We introduce Content-aware KG-enhanced Meta-preference Networks that enhances the CF recommendation based on both structured information from KG as well as unstructured content features based on Transformer-empowered content-based filtering (CBF). Within this modeling framework, we demonstrate a powerful KG-based CF model and a CBF model (a variant of the well-known NRMS system) and employ a novel training scheme, Cross-System Contrastive Learning, to address the inconsistency of the two very different systems in fusing information. We present experimental results showing that enhancing collaborative filtering with Transformer-based features derived from content-based filtering offers new improvements relative to strong baseline systems, improving the ability of KG-based CF systems to exploit item content information.

Keywords

Knowledge graph, recommender systems, collaborative filtering

1. Introduction

Collaborative Filtering (**CF**) and *Content-based Filtering* (**CBF**) are two leading recommendation techniques [1]. CF systems study users' interactions in order to leverage inter-item, inter-user, or user-item dependencies in making recommendations. The underlying notion is that users who interact with similar sets of items are likely to share preferences for other items. CBF models leverage descriptive attributes of items (e.g. item description and category) and users (e.g. age and gender). Users are characterized by the content information available in their browsing histories [2]. CBF is particularly well-suited to news recommendations, where millions of new items are produced every day. In contrast, CF systems are better suited to scenarios where the inventory of items grows slowly and where abundant user-item interactions are available. Movie and book recommender systems are examples of such scenarios and serve as the focus of this paper.

In the Netflix Prize competition (2006-2009) [3], CF features (ratings and user-item interactions) were shown to be more valuable than CBF features (e.g. movies' metadata) in recommendation [4]. However, recent work has shown that CF systems can benefit from the incorporation of external knowledge graphs (KGs) to enrich the user/item representations with *structured* CBF features [5]. Knowledge graphs consist of knowledge triplets; each triplet has a head entity, a tail entity, and a link that describes their relationship, e.g. [Christopher Nolan] - [director] - [Dunkirk (movie)]. KG-based CF models are particularly good at linking items to other related knowledge graph entities that serve as "item properties". This approach leverages the structured content information from KGs (e.g. movie genre and actors) to complement CF features.

While KGs can readily incorporate structured content information and external knowledge, *unstructured* content such as item descriptions, is largely unexploited. Recent Transformer-based models, such as BERT [6] and GPT-2 [7], have shown great power in modeling descriptive content from natural language, which offers new opportunities to enrich item/user representations with more expressive CBF features derived from Transformers. For example, the two movies "Interstellar" and "Inception", have a very similar set of structured properties in-

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)

cluding 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.

Therefore, in this work, we offer insights into the complementary power of unstructured CBF features derived from Transformers (e.g. summary texts of books and movies). We investigate how these content-aware CBF features can be effectively fused to complement CF learning, and how much value they can add to standard large-scale KG-based CF recommender systems.

However, computationally efficient approaches to enrich KG-based CF models with unstructured CBF features derived from Transformers are not yet well addressed in the literature. The challenge mainly stems from the need to capture the co-occurrence of graph node features by graph convolution operations. This operation requires representations of graph nodes to be back-propagated and updated after each forward pass, and thus it is prohibitively costly for large graphs where millions of item/user nodes require transformer-generated embeddings. Therefore, using pre-extracted features from trained CBF systems is the most promising option. However, conventional fusion schemes (such as Mixture of Expert and early/late fusion) are shown to be vulnerable in our experiments (see Sec. 4.4). We address this problem by introducing Cross-System Contrastive Learning, which brings together the benefits of both structured and unstructured item properties. In this paper:

1. We introduce a powerful KG CF model (KMPN) that outperforms strong baselines, and demonstrate the improvement brought by each system component. We also introduce a Transformer-empowered CBF model (NRMS-BERT) that achieves good recommendation performance with only summary texts of books and movies.
2. We propose to merge unstructured content-based features into KG-based CF through a simple but effective fusion framework based on Cross-System Contrastive Learning.
3. Based on two realistic recommendation datasets, we present extensive experiments showing the value of incorporating unstructured CBF features derived from Transformers.

2. Related Work

Collaborative Filtering. Traditional CF models rely on Matrix Factorization (MF) [8, 9, 10] and Factorization Machine (FM) [11, 12, 13] in learning user-item representations. Nearest neighbour approaches are also predominant to CF, where the user-item ratings are interpolated from the ratings of similar items and users [14, 15, 16]. Recent models incorporate Deep Neural Networks (DNN) in learning [17, 12, 18, 19, 20, 21]. Building upon graph-based CF models [22, 23], KG-based CF models fuse external knowledge from auxiliary KGs to improve both the accuracy and explainability of recommendation [5, 24]. Items in interaction graphs are associated with auxiliary KG entities with respect to their attributes (e.g. movie directors).

To exploit the KGs, *Embedding-based Methods* employ KG embedding methods (e.g TransE [25], TransH [26] and TransR [27]) in order to enhance item representations with KG-aware entity embeddings [28, 29, 30]. For example, KTUP [30] trains item representations and TransH-powered KG completion simultaneously. *Path-based Methods* follow the meta-paths manually designed by domain experts to make KG-path-aware recommendations [31, 32, 33, 34], which is, however, not feasible for larger KGs with their enormous entity and path diversity. *Convolution Methods* [35, 36, 32, 37, 38] design convolution mechanisms, mostly variants of Graph Neural Networks [39, 40] (GNNs), to enhance item/user representations with features aggregated from distant entities. KGIN [41] further embeds KG-relational embeddings in inter-node feature passing to achieve path-aware graph convolution.

Content-based Filtering. CBF models match items to a user by considering the metadata (content-based information) of items with which the user has interacted [42, 43, 44, 45, 46]. 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 [47, 48]. However, these methods heavily rely on word-level entity mapping with KG entities, which is prohibited for movies/books since their descriptions mostly consist of imaginary content, such as character names and fictional stories.

Fusing CF and CBF. Hybrid CF-CBF systems are often achieved by weighting/combining [49, 50] or switching [51, 52, 53] between the ranking outputs of the two systems. They can also pass a relatively coarser ranking list produced by one system into the other for refinement [54, 55]. The features derived from one system can also be used to complement the other system by fusing with the output features (late fusion) [56] or augmenting the user/item input features (early fusion) [57, 58]. For example, CKE [29] produces augmented item representations by obtaining fixed textual features from unsupervised denoising auto-encoders. In contrast, we introduce NRMS-BERT to obtain more expressive textual item representations with supervised training and larger language models. Furthermore, these conventional fusing approaches (including late/early fusion and mixture of experts) fail to perform well in our experiments (Sec. 4.4). We address this by proposing a novel training scheme based on contrastive learning that complements a KG-

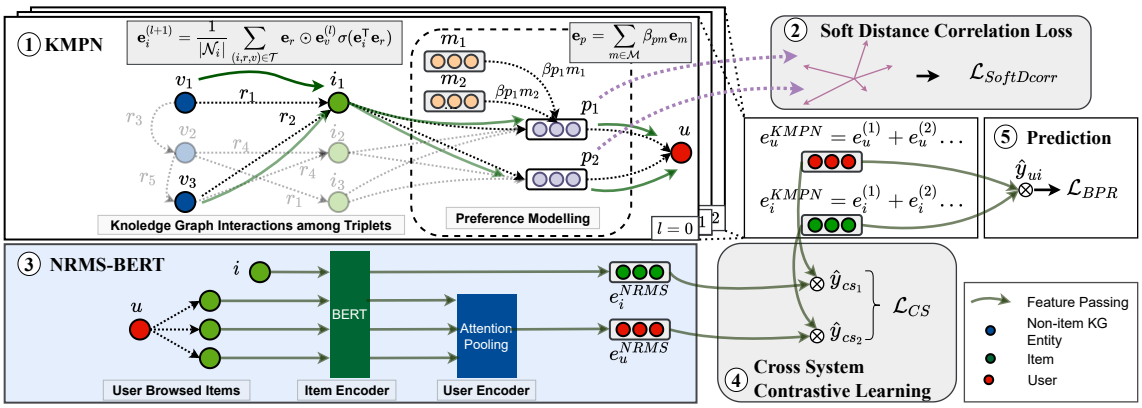


Figure 1: Framework pipeline. (1) KMPN: leverages meta preferences to model users from knowledge graph entities and interacted items; (2) Soft Distance Correlation: encourage preference embeddings to separate at low dimensions; (3) NRMS-BERT: extracts content-based features; (4) Cross System Contrastive Learning: encourages user/item embeddings to learn mutual information from content-based representations; (5) Rating: uses the dot product of KMPN user/item features.

based CF model with these Transformer-based representations.

3. Methodology

3.1. Data Notation

There are N_u users $\{u|u \in \mathcal{S}_U\}$ and N_i items $\{i|i \in \mathcal{S}_I\}$. $\mathcal{P}^+ = \{(u,i)|u \in \mathcal{S}_U, i \in \mathcal{S}_I\}$ is the set of user interactions. Each (u,i) pair indicates that user u interacted with i. Each item $i \in \mathcal{S}_I$ carries unstructured data $\mathbf{x}_i$, e.g. a text description of the item.

The KG contains structured information that describes relations between real world entities. The KG is represented as a weighted heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node set $\mathcal{V}$ consisting of N_v nodes $\{v\}$ and an edge set $\mathcal{E}$ containing all edges between nodes. The graph is also associated with a relation type mapping function $\phi : \mathcal{E} \rightarrow \mathcal{R}$ that maps each edge to a type in the relation set $\mathcal{R}$ consisting of N_r relations. Note that all items are included in the KG: $\mathcal{S}_i \subset \mathcal{V}$.

The edges of the knowledge graph are triplets $\mathcal{T} = \{(h,r,t)|h,t \in \mathcal{V}, r \in \mathcal{R}\}$, where $\mathcal{V}$ is the collection of graph entities/nodes and $\mathcal{R}$ is the relation set. Each triplet describes that a head entity h is connected to a tail entity t with the relation r. For example, $(The\ Shining, film.film.actor, Jack\ Nicholson)$ specifies that Jack Nicholson is a film actor in the movie "The Shining". To fully expose the relationships between heads and tails, the relation set is extended with reversed relation types, i.e., for any (h,r,t) triplet we allow inverse connection (t,r',h) to be built, where r' is the reverse of r. The edge set $\mathcal{E}$ is derived from these triplets.

3.2. KG-Enhanced Meta-Preference Network (KMPN)

This section introduces KG-enhanced Meta-Preference Network (thereafter KMPN). It is a KG-based CF model that aggregates features of all KG entities to items efficiently by exploiting relationships in the KG, and then links item features to users for recommendations, as shown in Fig. 1 (1), (2), and (5).

3.2.1. Gated Path Graph Convolution Networks

Associated with each KG node v_i are feature vectors $\mathbf{e}_i^{(0)} \in \mathbf{R}^h$. Each relation type $r \in \mathcal{R}$ is also associated with a relational embedding $\mathbf{e}_r$. A Gated Path Graph Convolution Network is a cascade of L convolution layers. For each KG node, a convolution layer aggregates features from its neighbors as follows:

$$\mathbf{e}_i^{(l+1)} = \frac{1}{|\mathcal{N}_i|} \sum_{\{v_j|(v_i,r_{ij},v_j)\in\mathcal{T}\}} \gamma_{ij}\mathbf{e}_{r_{ij}} \odot \mathbf{e}_j^{(l)}, \quad (1)$$

where the neighbouring set of i: $\mathcal{N}_i = \{v_j|(v_i,r_{ij},v_j) \in \mathcal{T}\}$, r_{ij} is the type of relation from v_i to v_j, and γ_{ij} is a gated function that controls messages that flow from v_j to v_i:

$$\gamma_{ij} = \sigma(\mathbf{e}_i^\top \mathbf{e}_{r_{ij}}), \quad (2)$$

where $\sigma(.)$ is a sigmoid function that limits the gated value between 0 and 1. As a result, the message passed to a node is weighted by its importance to the receiving node and the relation type. Through stacking multiple layers of convolution, the final embedding at a node depends on the path along which the features are shared, as well as the importance of the message being transmitted. To overcome the over-smoothing issue of graph

convolutions, the embedding at a KG node after l convolutions is an aggregation of all the intermediate output embeddings: $\mathbf{e}_i^l = \sum_{l'=0}^{l} \mathbf{e}_i^{(l')}$.

3.2.2. User Preference Modeling

Inspired by Wang et al. [41], we model users using a combination of preferences. Wang et al. [41] assumed that each user is influenced by multiple intents and that each intent is influenced by multiple movie attributes, such as the combination of the two relation types $film.film.director$ and $film.film.genre$. Based on this assumption, they proposed to aggregate item embeddings to users through "preferences", and the embedding of each preference $\mathbf{e}_p$ is modelled by all types of edges: $\mathbf{e}_p = \sum_{r \in \mathscr{R}} \alpha_{rp} \mathbf{e}_r$, where α_{rp} is a Softmax-ed trainable weight and $\mathbf{e}_r$ is the embedding of edge relation type r.

We take the view that user preferences are not only limited to relations but can be extended to more general cases. We model each preference p through a combination of a set of meta-preferences $\mathscr{M}$ with in total N_m meta-preferences: each meta-preference $m \in \mathscr{M}$ is associated with a trainable embedding $\mathbf{e}_m \in \mathbf{R}^h$, and a preference p is formed by these meta-preferences as follows:

$$\mathbf{e}_p = \sum_{m \in \mathscr{M}} \beta_{pm} \mathbf{e}_m, \tag{3}$$

where the linear weights $\{\beta_{pm} | m \in \mathscr{M}\}$ are derived from trainable weights $\{\hat{\beta}_{pm} | m \in \mathscr{M}\}$ for each preference p:

$$\beta_{pm} = \frac{\exp(\hat{\beta}_{pm})}{\sum_{m' \in \mathscr{M}} \exp(\hat{\beta}_{pm'})}. \tag{4}$$

As a result, meta-preferences reflect the general interests of all users. A particular user can be profiled by aggregating the embeddings of interacted items through these preferences:

$$\mathbf{e}_u^{(l+1)} = \sum_{p \in \mathscr{P}} \alpha_p \sum_{(u,i) \in \mathscr{P}^+} \mathbf{e}_i^{(l)} \odot \mathbf{e}_p, \tag{5}$$

where $\mathscr{P}$ is the collection of N_p preferences $\{p\}$ and α_p is an attention mechanism that weights the interest of users over different preferences:

$$\alpha_p = \frac{\exp(\mathbf{e}_p^\mathsf{T} \mathbf{e}_u^{(l)})}{\sum_{p' \in \mathscr{P}} \exp(\mathbf{e}_{p'}^\mathsf{T} \mathbf{e}_u^{(l)})}. \tag{6}$$

In summary, each preference is formed by general and diverse meta-preferences, and users are further profiled by multiple preferences that focus on different aspects of item features. As with items, the final user embedding is: $\mathbf{e}_u^l = \sum_{l'=0}^{l} \mathbf{e}_u^{(l')}$.

3.2.3. Soft Distance Correlation

Having modelled users through preferences, Wang et al. [41] added an additional loss that utilizes Distance Correlation (DCorr) [59, 60] to separate the representations of these learnt preferences as much as possible, in order to obtain diverse proxies bridging users and items. Though the authors demonstrate a considerable improvement over baselines, we take the view that applying constraints to all dimensions of preference embeddings restricts their expressiveness, as they are trained to be very dissimilar and have diverse orientations in latent space.

We adopt a softer approach: **Soft Distance Correlation Loss**, which firstly lowers the dimensionality of preference embeddings with *Principal Component Analysis* (PCA) [61] while keeping the most differentiable features in embeddings, and then applies distance correlation constraints to encourage diverse expression in lower dimensions:

$$\hat{\mathbf{e}}_p = PCA(\{\mathbf{e}_{p'} | p' \in \mathscr{P}\}) \in \mathbf{R}^{h\epsilon}; \tag{7}$$

$$\mathscr{L}_{SoftDCorr} = \sum_{p,p' \in \mathscr{P}, p \neq p'} \frac{DCov(\hat{\mathbf{e}}_p, \hat{\mathbf{e}}_{p'})}{\sqrt{DVar(\hat{\mathbf{e}}_p) \cdot DVar(\hat{\mathbf{e}}_{p'})}}. \tag{8}$$

where ϵ controls the ratio of principal components to keep after PCA. $DCov(\cdot)$ computes distance covariance and $DVar(\cdot)$ measures distance variance [59, 60].

Of course, $\epsilon = 1$ yields the original DCorr Loss proposed in [41]. Through encouraging diverse expression at only lower dimensions, preferences have retained the flexibility in higher dimensions.

3.2.4. Model Optimization with Reciprocal Ratio Negative Sampling (RRNS)

Following the common approach, the dot product between user and item embeddings is used for rating: $\hat{y}_{ui} = (\mathbf{e}_u^L)^\mathsf{T} \cdot \mathbf{e}_i^L$.

Both of the datasets we study do not provide hard negative samples: i.e., we do not have samples of items with which viewers chose not to interact. A common practice to synthesize negative examples is to randomly sample from users' unobserved counterparts $\mathscr{P}^- = \{(u, i^-) | (u, i^-) \notin \mathscr{P}^+\}$. However, an item is not necessarily "not interesting" to a user if no interaction happens, as not all items have been viewed. We propose to adopt *Reciprocal Ratio Negative Sampling* (RRNS), where items with more user interactions are considered popular and are sampled less frequently based on the assumption that popular items are less likely to be hard negative samples for any user. The sampling distribution is given by a normalized reciprocal ratio of item interactions:

$$i^- \sim P(i) \propto \frac{1}{c(i)} \text{ for } i \in S_I \tag{9}$$

where $c(i)$ counts the interactions of all users with the item i.

The training set therefore consists of positive and negative samples: $\mathscr{U} = \{(u, i^+, i^-) | (u, i^+) \in \mathscr{P}^+, (u, i^-) \in \mathscr{P}^-\}$. Pairwise BPR loss [9] is adopted to train the model, which

exploits a contrastive learning concept to assign higher scores to users' browsed items than those items in which the users are not interested:

$$\mathscr{L}_{BPR} = \sum_{(u,i^+,i^-)\in\mathscr{U}} -\ln(\sigma(\hat{y}_{ui^+} - \hat{y}_{ui^-})). \tag{10}$$

Together with commonly-used embedding L2 regularization and Soft Distance Correlation loss, the final loss is given by:

$$\mathscr{L}_{KMPN} = \mathscr{L}_{BPR} + \lambda_1 \frac{1}{2}\|\Theta\|_2^2 + \lambda_2 \mathscr{L}_{SoftDCorr}, \tag{11}$$

where $\Theta = \{\mathbf{e}_u^L, \mathbf{e}_{i^+}^L, \mathbf{e}_{i^-}^L | (u, i^+, i^-) \in \mathscr{U}\}$, and $\|\Theta\|_2^2$ is the L2-norm of user/item embeddings. λ_1 and λ_2 are hyperparameters that control loss weights.

3.3. Neural Recommendation with Multi-Head Self-Attention

Inspired by NRMS [43] that is powerful in news recommendations, we propose a variant of NRMS, **NRMS-BERT**, that further utilizes a fine-tuned Transformer (BERT) for extracting contextual information from descriptions of items, as shown in Fig. 1 (3).

3.3.1. Item Encoder

The item encoder encodes the text description string $\mathbf{x}_i$ of any item $i \in \mathscr{S}_i$ through BERT into embeddings of size h by extracting the embeddings of <CLS> at the last layer:

$$\mathbf{e}_i = BERT(\mathbf{x}_i) \in \mathbf{R}^h. \tag{12}$$

For each user u, the item encoder encodes one positive item $\mathbf{e}_{i^+}$ and K negative items $\mathbf{e}_{i_1^-}$, ..., $\mathbf{e}_{i_K^-}$. B items are randomly sampled from the user's browsed items $i_{u,1}$,...,$i_{u,B}$. These browsed items are encoded and gathered to $\mathbf{E}_u = [\mathbf{e}_{i_{u,1}}, ..., \mathbf{e}_{i_{u,B}}] \in \mathbf{R}^{B\times h}$.

3.3.2. User Encoder

The user encoder uses items with which users interacted to produce a content-aware user representation. The final user representation is a weighted sum of the B browsed items:

$$\mathbf{e}_u = \sum_{b=1}^{B} \alpha_b \mathbf{e}_{i_{u,b}} \tag{13}$$

where α_b is the attention weight assigned to $i_{u,b}$ obtained by passing features through two linear layers:

$$\alpha_b = \frac{\exp(\hat{\mathbf{A}}_b)}{\sum_{b'=1,..,B}\exp(\hat{\mathbf{A}}_{b'})}; \tag{14}$$

$$\hat{\mathbf{A}} = \tanh(\mathbf{E}_u\mathbf{A}_{fc_1} + \mathbf{b}_{fc_1})\mathbf{A}_{fc_2} + \mathbf{b}_{fc_2} \in \mathbf{R}^{B\times 1} \tag{15}$$

where $\mathbf{A}_{fc_1} \in \mathbf{R}^{h\times\frac{1}{2}h}$, $\mathbf{b}_{fc_1} \in \mathbf{R}^{\frac{1}{2}h}$, $\mathbf{A}_{fc_2} \in \mathbf{R}^{\frac{1}{2}h\times 1}$, and $\mathbf{b}_{fc_2} \in \mathbf{R}^1$ are weights and biases of two fully-connected layers, respectively.

3.3.3. Model Optimization

The rating is the dot product of user and item embeddings: $\hat{y}_{ui} = (\mathbf{e}_u)^\mathsf{T} \cdot \mathbf{e}_i$. Assume that the scores of the positive samples and negative samples are $\hat{y}^+$ and $\hat{y}_1^-$,...,$\hat{y}_K^-$, following [43], the loss is the log click probability of item i:

$$\mathscr{L}_{NRMS} = -\sum_{i\in\mathscr{S}_i} \log\left(\frac{\exp(\hat{y}^+)}{\exp(\hat{y}^+) + \sum_{k=1,..,K}\exp(\hat{y}_k^-)}\right) \tag{16}$$

3.4. Fusing CF and CBF: Content-aware KMPN (CKMPN)

To fuse the information from a CBF model (NRMS-BERT) to a CF model (KMPN), we must bridge some inconsistencies between the two types of models. CBF models that utilize large transformers cannot be co-optimized with KG-based CF models, as graph convolution requires all embeddings to be present before convolution and this requires enormous GPU memory for even one single forward pass. As a result, a more efficient solution merges the pre-trained CBF features into the training of the KG-CF component, enriching the learned representations.

In line with our aim to use a CF model for movie and book recommendations, we present a novel and efficient approach for training a better KMPN: *Cross-System Contrastive Learning*, as shown in Fig. 1 (4). KMPN is still used as the backbone and it is trained with the aid of a pre-trained NRMS-BERT, not requiring more parameters than KMPN.

In KMPN training, for users and items in $(u, i^+, i^-) \in \mathscr{U}$, embeddings are generated from NRMS-BERT: e_u^{NRMS}, $e_{i^+}^{NRMS}$, $e_{i^-}^{NRMS}$, and from KMPN: e_u^{KMPN}, $e_{i^+}^{KMPN}$, and $e_{i^-}^{KMPN}$.

Cross-System Contrastive Loss is adopted to encourage the KMPN system to learn to incorporate content-sensitive features from NRMS-BERT features:

$$\begin{aligned}\mathscr{L}_{CS} = \sum_{(u,i^+,i^-)\in\mathscr{U}} &-\ln\left(\sigma((\mathbf{e}_u^{KMPN})^\mathsf{T}\cdot(\mathbf{e}_{i^+}^{NRMS} - \mathbf{e}_{i^-}^{NRMS}))\right) \\ &-\ln\left(\sigma((\mathbf{e}_u^{NRMS})^\mathsf{T}\cdot(\mathbf{e}_{i^+}^{KMPN} - \mathbf{e}_{i^-}^{KMPN}))\right)\end{aligned} \tag{17}$$

This loss encourages KMPN to produce item embeddings that interact not only with KMPN's own user embeddings, but also with NRMS-BERT's user embeddings. Similarly, user embeddings of KMPN are trained to interact with items of NRMS-BERT. This allows $\mathbf{e}_i^{KMPN}$ to learn mutual expressiveness with $\mathbf{e}_i^{NRMS}$, but without approaching the two embeddings directly using similarity (e.g. cosine-similarity), which we found not to work well (discussed in Sec. 4.4). In this case, $\mathbf{e}_u^{NRMS}$ serves as an 'anchor' with which the item embeddings of two systems learn to share commons and increase their mutuality. This loss encourages $\mathbf{e}_i^{KMPN}$ and $\mathbf{e}_i^{NRMS}$ to lie in the

Table 1
Model performance on Amazon-Book-Extended (top) and Movie-KG-dataset (bottom). Numbers underlined represent existing state-of-the-art performance, while best performance of the proposed models is marked in **bold**. The average of 3 runs is reported to mitigate experimental randomness. Metrics with (*) are significantly higher than KMPN ($p < 0.05$).

On Amazon-Book-Extended	Recall			ndcg			Hit Ratio		
	@20	@60	@100	@20	@60	@100	@20	@60	@100
BPRMF	0.1352	0.2433	0.3088	0.0696	0.0957	0.1089	0.2376	0.3984	0.4816
CKE	0.1347	0.2413	0.3070	0.0691	0.0948	0.1081	0.2373	0.3963	0.4800
KGAT	0.1527	0.2595	0.3227	0.0807	0.1066	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 (ours)	**0.1719**	0.2793	0.3405	**0.0931**	0.1189	0.1315	**0.2910**	0.4421	0.5166
- w/o Soft DCrr	0.1704	0.2790	0.3396	0.0924	0.1185	0.1310	0.2881	0.4419	0.5152
- w/o Soft DCorr and RRNS	0.1690	0.2774	0.3391	0.0913	0.1177	0.1302	0.2872	0.4414	0.5155
NRMS-BERT (ours)	0.1142	0.2083	0.2671	0.0592	0.0817	0.0935	0.2057	0.3487	0.4273
CKMPN ($\lambda_{CS} = 0.2$) (ours)	0.1699	0.2812	**0.3461**	0.0922	0.1190	0.1319	0.2880	0.4460	0.5235
CKMPN ($\lambda_{CS} = 0.1$) (ours)	0.1718	**0.2821***	0.3460*	0.0928	**0.1197***	**0.1326***	0.2908	**0.4474***	**0.5244***
Improv. (%) CKMPN v.s. Best Baselines	3.90	4.82	4.94	4.31	4.55	4.59	3.72	4.33	4.04

On Movie-KG-Dataset	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 ($\epsilon = 0.5, N_m = 64$) (ours)	0.1434	0.2130	0.2427	0.1073	0.1305	0.1367	0.2193	0.3098	0.3602
NRMS-BERT (ours)	0.1241	0.1669	0.1890	0.1034	0.1213	0.1257	0.1728	0.2369	0.2773
CKMPN ($\lambda_{CS} = 0.01$) (ours)	**0.1457**	**0.2157**	**0.2462**	**0.1149**	**0.1417**	**0.1482**	**0.2266**	**0.3153**	**0.3668**
CKMPN (ours) (on the cold-start set)	0.1024	0.1741	0.2130	0.0570	0.0729	0.0808	0.1812	0.2839	0.3380

same hidden space hyperplane on which features have the same dot-product results with $\mathbf{e}_u^{NRMS}$. This constraint encourages KMPN to grow embeddings in the same region of hidden space, leading to mutual expressiveness across the two systems. Finally, the optimization target is:

$$\mathscr{L}_{CKMPN} = \mathscr{L}_{KMPN} + \lambda_{CS}\mathscr{L}_{CS}, \tag{18}$$

where λ_{CS} controls the weight of the Cross-System Contrastive Loss. This fusion scheme can be applied to any models with similar CF/CBF mechanisms.

4. Experiments

4.1. Datasets

We use the two datasets introduced in [62]: (1) Amazon-Book-Extended collects book descriptions from multiple data sources for the popular Amazon-Book dataset. It contains 70,679 users, 24,915 items along with a KG of 88,572 nodes and 2,557,746 triplets. (2) Movie-KG-Dataset is a newly collected dataset that contains 125,218 users, 50,000 items with a KG of 250,327 nodes and 12,055,581 triplets. Descriptions of movies are provided to enable content-based recommendations.

4.2. 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 [63] 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. Training takes 4 hours on Amazon-Book-Extended and 12 hours on Movie-KG-Dataset. NRMS-BERT is trained for 10 epochs at a constant learning rate of 10^{-4}. Training takes 20 hours on Amazon-Book-Extended and 120 hours on Movie-KG-Dataset.

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

4.3. Evaluation Metrics and Baselines

Following common practice [21, 37, 41, 64], 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) [64]: increases when relevant items appear earlier in the rec-

ommended list; (3) *HitRatio@K*: how likely a user finds at least one interesting item in the recommended top-K items.

We take the performance of several recently published recommender systems as points for comparison[1]. We carefully reproduced all these baseline systems from their repositories[2].

BPRMF [9]: a strong Matrix Factorization (MF) method that applies a generic optimization criterion BPR-Opt for personalized ranking. Limited by space, other MF models (e.g. FM [65], NFM [12]) are not presented since BPRMF outperformed them.

CKE [29]: a CF model that leverages heterogeneous information in a knowledge base for recommendation.

KGAT [37]: 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 [41]: a state-of-the-art KG-based CF model that models users' latent intents (preferences) as a combination of KG relations.

4.4. Performance on Amazon Dataset

Comparison with baselines. Performance of models is presented in Table 1. Our proposed KG-based CF model, KMPN, achieved a substantial improvement on all metrics over the performance of the existing state-of-the-art model KGIN; for example, Recall@20 was improved from 0.1654 to 0.1719, Recall@100 from 0.3298 to 0.3405, and ndcg@100 from 0.1267 to 0.1315. **All relative improvements mentioned in our discussions are statistically significant ($p < 0.05$).**

NRMS-BERT models user-item preferences using only item summary texts, without external information from a knowledge base. It still achieves 0.1142 in Recall@20 and 0.4273 Hit Ratio@100, not far from the KGIN baseline at 0.5040 Hit Ratio@100.

CKMPN further improves all @60/@100 metrics while keeping the model's performance of @20. For example, with similar Recall@20, CKMPN (0.3461 Recall@100) outperforms KMPN (0.3405 Recall@100) by 1.6% with statistical significance $p < 0.05$. This demonstrates that even though KMPN achieves higher performance relative to NRMS-BERT, gathering item and user embeddings from one system (KMPN) with those of the other system (NRMS-BERT), through proxies (Cross-System CL), can still encourage KMPN to learn and fuse content-aware information from the learned representations of a CBF model and presents more relevant items in the top-100 list.

[1]They are also baseline systems being compared in a recent paper [41] (WWW'21).

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

Comparison with hybrid methods: Conventional feature fusion methods are popular and convenient options for combining one system into the training of another (as surveyed in Sec. 2). In fusing a pre-trained NRMS-BERT with KMPN, we demonstrate the effectiveness of our proposed fusion framework CKMPN by comparing it with these conventional approaches.

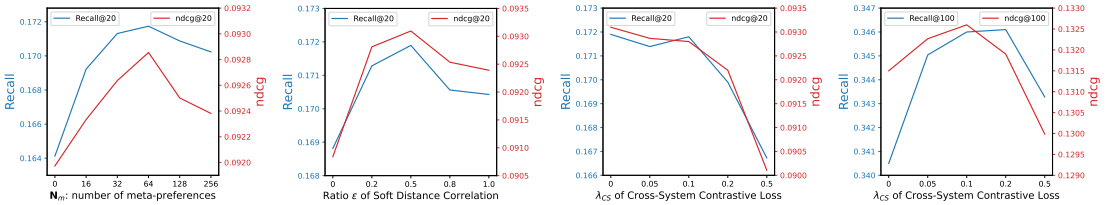
- **Early Fusion**: CBF features are concatenated to the trainable user/item embeddings of KMPN before the graph convolution layers.
- **Late Fusion**: CBF features are fused to the output user/item embeddings of KMPN after the graph convolution layers. Many feature aggregation methods were experimented and the best of them are reported in Table 2: (1) concat+linear: CF features are concatenated with CBF features, and they pass through 3 MLP layers into embeddings of size $\mathbf{R}^{2\times h}$. (2) MultiHeadAtt: CF and CBF features passed through 3 Multi-head Self-Attention blocks into embeddings of size $\mathbf{R}^{2\times h}$.
- **Cos-Sim**: An auxiliary loss grounded on cosine-similarity is incorporated in training to encourage the user/item embeddings of KMPN to approach those of NRMS-BERT.
- **Mixture of Expert (MoE)**: a hybrid system where the output scores of two systems, KMPN and NRMS-BERT, pass through 3 layers of a Multi-Layer Perception (MLP) to obtain final item ratings.

It can be concluded that these feature aggregation approaches do not perform well in fusing pre-trained CBF features into KG-based CF training. (1) The performance of **Late Fusion** shows that when the already-learned NRMS-BERT item/user embeddings pass through new layers, these layers undo the learned representations from NRMS-BERT and lead to only degraded performance. (2) **Cos-Sim** shows that the auxiliary loss based on cosine-similarity places a reliance on NRMS-BERT's features, which damages the KMPN training by limiting the expressiveness of KMPN to that of NRMS-BERT. As a result, the performance is decreased from 0.2793 (KMPN) to 0.2436 (Cos-Sim) recall@60.

Though NRMS-BERT alone achieves much lower metrics than KMPN (0.1142 vs 0.1719 Recall@20), **MoE**, where scores of two systems are merged by MLP layers, achieves 0.1723 Recall@20, showing that the scoring of two systems is complementary. However, MoE's performance deteriorates on @60/100. A case study is presented later in Sec. 4.6 to show that the scoring of one system can possibly be extreme to overwhelm the final rating under the MoE setting. In contrast, our CKMPN steadily achieves better performance in @60/100 results relative to KMPN, showing that our method is an in-depth collaboration of two systems instead of a simple aggregation of system outputs as in MoE.

Table 2
Comparison with conventional feature fusion approaches (Amazon-Book-Extended). R: Recall; HR: Hit Ratio.

Fusion Approach	R@20	R@60	ndcg@60	HR@60
Early Fusion (concat)	0.1661	0.2708	0.1148	0.4299
Late Fusion (concat+linear)	0.1679	0.2769	0.1164	0.4381
Late Fusion (MultiHeadAtt)	0.1692	0.2778	0.1175	0.4385
Cos-Sim	0.1436	0.2436	0.1026	0.4001
Mixture of Expert	**0.1723**	0.2791	0.1161	0.4425
CKMPN (ours)	0.1718	**0.2821**	**0.1197**	**0.4474**



(a) Model performance varies with the number of meta-preferences.
(b) Model performance varies with the ratio ϵ of Soft Distance Correlation (DCorr).
(c) Recall@20 (blue) and ndcg @20 (red) against the loss weight λ_{CS}.
(d) Recall@100 (blue) and ndcg @100 (red) against the loss weight λ_{CS}.

Figure 2: Evaluation of model hyperparameters. Zoom in to see figures in detail.

In conclusion, Cross-System CL significantly enhances KMPN's ability to present more relevant items in the top-100 list through the fusion of unstructured content-based features. It complements the aforementioned shortages of conventional fusing methods by merging features without corrupting the already-learned representation and without directly approaching two systems' outputs.

4.5. Contributions of Components

To support the rationale of our designs, Ablation studies and hyperparameter evaluation are presented to explore the effects of each proposed component.

Effects of Meta Preferences. An important research question is how the design of modeling users through meta-preferences improves the model performance. As shown in Fig. 2a, removing meta-preference modeling of users from KMPN ($N_m = 0$) dramatically decreases the performance, showing that modeling users' preferences is necessary. $N_m = 16$ achieves worse performance than $N_m \geq 32$ since a small number of meta preferences limits the model's capacity of modeling users. The performance on all metrics increases until it peaks at $N_m = 64$, and then it starts to decrease at $N_m \geq 128$. This suggests that including too many meta preferences induces overfitting and does not further improve the system. It is a good model property in practice since a moderate $N_m = 64$ is sufficient for achieving the best performance.

Effects of Soft Distance Correlation Loss. The hyperparameter ϵ controls the number of principal components to keep after PCA dimension reduction. The lower the ratio, the more flexibility the preference embeddings will recover in dimensions from the standard Distance Correlation (DCorr) constraint. As shown in Fig. 2b, $\epsilon = 0$ (left) removes the DCorr constraint completely, while $\epsilon = 1$ (right) reduces to a standard DCorr Loss. As ϵ approaches 0, the DCorr constraint becomes too loose to encourage the diversity of preferences, leading to a dramatically decreased performance. The performance peaks at $\epsilon = 0.5$, where half of h dimensions are relaxed from the standard Dcorr constraint, and preference embeddings are still able to grow diversely in the remaining half dimensions. This suggests that our softer version of DCorr constraint is beneficial to user modeling.

Effects of RRNS. As shown in Table 1, without *Reciprocal Ratio Negative Sampling*, Recall@20 of KMPN (w/o SoftDcorr) is decreased from 0.1704 to 0.1690. In line with our intuition, reducing the probability of sampling popular items as negative samples for training can yield benefits in model learning. This demonstrates that while viewed-but-not-clicked (hard negative) samples are not available to the model, our proposed sampling strategy enhances the quality of negative samples.

Effects of Cross-System Contrastive Learning. The system performance of top-20 does not drop much for $\lambda_{CS} \leq 0.2$ (Fig. 2c) whereas the performance at top-100 increases dramatically for $\lambda_{CS} \leq 0.2$ relative to a system without Cross-System CL ($\lambda_{CS} = 0$) (Fig. 2d). This suggests that by incorporating Cross-System CL in our training with a reasonable λ_{CS}, CKMPN is more capable of finding relevant items for users.

4.6. Performance on Movie-KG-Dataset

As shown in Table 1(bottom), the same performance boost is observed in KMPN relative to baselines. For example, KMPN achieves 0.1434 Recall@20 and 0.1073 ndcg@20, which is higher than 0.1403 Recall@20 and 0.1006 ndcg@20 of the baselines. CKMPN also achieves the best performance by incorporating content-based features from NRMS-BERT. It outperforms KMPN in all metrics, showing a significant improvement in ndcg@100 (from 0.1367 to 0.1482) and Hit Ratio@100 (from 0.3602 to 0.3668) in particular. Therefore, we can conclude that our method is applicable in multiple different datasets.

Table 3
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. NRMS: NRMS-BERT; MoE: Mixture of Expert.

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

An example output of systems is presented in Table 3. 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 Cross-System CL. 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 our Cross-System CL approach is an effective in-depth collaboration of two systems, outperforming the direct mixture of KMPN and NRMS-BERT.

We also present the model performance on the cold-start test set of the Movie-KG-dataset, where users are completely unseen in the training. As shown in the last section of Table 1 (bottom), our best model CKMPN still achieved good performance for unseen users on all metrics, e.g., 0.1024 on Recall@20 and 0.3380 on Hit Ratio@100. The performance did not deteriorate much from the standard test set, showing that our model still functions in the cold-start setting.

5. Conclusion

We present KMPN, a powerful KG-based CF model that outperforms strong baseline models. To investigate the complementary power of unstructured content-based information, we further propose a novel approach *Cross-System Contrastive Learning*, which combines CF and CBF, two distinct paradigms to achieve a substantial improvement relative to models in literature. This suggests that KG-based CF models can benefit from the incorporation of unstructured content information derived from Transformers.

Our proposed CKMPN has thus far achieved substantial improvements on both datasets, especially on top-60/100 metrics. Industrial recommender systems usually follow a 2-step pipeline where a relatively large amount of items $K = 60, 100$ is firstly recalled by a Recall Model and then a Ranking Model is adopted to refine the list ranking. This improvement presents more relevant items in the relatively coarser Recall output, which is appealing to industrial applications. Also, CKMPN is much more preferred than the Mixture of Expert model in industrial applications, since it still produces independent user/item representations. This feature enables the fast and efficient match of users and items in hidden space with $O(log(n))$ query time complexity [66].

References

[1] G. Takács, I. Pilászy, B. Németh, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, The Journal of Machine Learning Research 10 (2009) 623–656.

[2] P. B. Thorat, R. Goudar, S. Barve, Survey on collaborative filtering, content-based filtering and hybrid recommendation system, International Journal of Computer Applications 110 (2015) 31–36.

[3] J. Bennett, S. Lanning, et al., The netflix prize, in: Proceedings of KDD cup and workshop, volume 2007, Citeseer, 2007, p. 35.

[4] I. Pilászy, D. Tikk, Recommending new movies: even a few ratings are more valuable than metadata, in: Proceedings of the third ACM conference on Recommender systems, 2009, pp. 93–100.

[5] 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).

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] K. Verstrepen, B. Goethals, Unifying nearest neighbors collaborative filtering, in: Proceedings of the 8th ACM Conference on Recommender systems, 2014, pp. 177–184.

[15] M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms, ACM Transactions on Information Systems - TOIS 22 (2004) 143–177. doi:10.1145/963770.963776.

[16] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: Proceedings of the 2nd ACM Conference on Electronic Commerce, 2000, pp. 158–167.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[26] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 28, 2014.

[27] Z. Wang, J. Li, Z. Liu, J. Tang, Text-enhanced rep-

resentation learning for knowledge graph, in: Proceedings of International Joint Conference on Artificial Intelligent (IJCAI), 2016, pp. 4–17.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] W. L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 1025–1035.

[40] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph Attention Networks, International Conference on Learning Representations (2018).

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] S. H. Choi, Y.-S. Jeong, M. K. Jeong, A hybrid recommendation method with reduced data for large-scale application, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40 (2010) 557–566.

[50] L. M. De Campos, J. M. Fernández-Luna, J. F. Huete, M. A. Rueda-Morales, Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks, International journal of approximate reasoning 51 (2010) 785–799.

[51] D. Billsus, M. J. Pazzani, J. Chen, A learning agent for wireless news access, in: Proceedings of the 5th international conference on Intelligent user interfaces, 2000, pp. 33–36.

[52] M. Ghazanfar, A. Prugel-Bennett, Building switching hybrid recommender system using machine learning classifiers and collaborative filtering, IAENG International Journal of Computer Science 37 (2010).

[53] J. M. Noguera, M. J. Barranco, R. J. Segura, L. Martínez, A mobile 3d-gis hybrid recommender system for tourism, Information Sciences 215 (2012) 37–52.

[54] A. S. Lampropoulos, P. S. Lampropoulou, G. A. Tsihrintzis, A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis, Multimedia Tools and Applications 59 (2012) 241–258.

[55] I. A. Christensen, S. N. Schiaffino, A hybrid approach for group profiling in recommender systems (2014).

[56] P. Bedi, P. Vashisth, P. Khurana, et al., Modeling user preferences in a hybrid recommender system using type-2 fuzzy sets, in: 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2013, pp. 1–8.

[57] R. J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, in: Proceedings of the fifth ACM conference on Digital libraries, 2000, pp. 195–204.

[58] X. Li, T. Murata, Multidimensional clustering based collaborative filtering approach for diversified recommendation, in: 2012 7th International Conference on Computer Science & Education (ICCSE), IEEE, 2012, pp. 905–910.

[59] G. J. Székely, M. L. Rizzo, Brownian distance covariance, The annals of applied statistics 3 (2009) 1236–1265.

[60] G. J. Székely, M. L. Rizzo, N. K. Bakirov, Measuring and testing dependence by correlation of distances, The annals of statistics 35 (2007) 2769–2794.

[61] H. Hotelling, Analysis of a complex of statistical variables into principal components., Journal of educational psychology 24 (1933) 417.

[62] W. Lin, L. Shou, M. Gong, P. Jian, Z. Wang, B. Byrne, D. Jiang, Combining unstructured content and knowledge graphs into recommendation datasets, in: 4th Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) Workshop @ RecSys 2022, 2022.

[63] 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.

[64] 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.

[65] 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.

[66] Y. A. Malkov, D. A. Yashunin, Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, IEEE transactions on pattern analysis and machine intelligence 42 (2018) 824–836.