

# Improving Content Recommendation: Knowledge Graph-Based Semantic Contrastive Learning for Diversity and Cold-Start Users

Yejin Kim<sup>1,2</sup>, Scott Rome<sup>2</sup>, Kevin Foley<sup>2</sup>, Mayur Nankani<sup>2</sup>,  
Rimon Melamed<sup>1</sup>, Javier Morales<sup>2</sup>, Abhay Yadav<sup>2</sup>, Maria Peifer<sup>2</sup>,  
Sardar Hamidian<sup>2†</sup> and H. Howie Huang<sup>1†</sup>

<sup>1</sup>GraphLab, George Washington University

<sup>2</sup>Applied AI Research, Comcast

{yeijnjenny,rmelamed,howie}@gwu.edu

{scott\_rome,kevin\_foley,javier\_moralesdelgado,abhay\_yadav3,maria\_peifer,sardar\_hamidian}@comcast.com

## Abstract

Addressing the challenges related to data sparsity, cold-start problems, and diversity in recommendation systems is both crucial and demanding. Many current solutions leverage knowledge graphs to tackle these issues by combining both item-based and user-item collaborative signals. A common trend in these approaches focuses on improving ranking performance at the cost of escalating model complexity, reducing diversity, and complicating the task. It is essential to provide recommendations that are both personalized and diverse, rather than solely relying on achieving high rank-based performance, such as Click-through rate, Recall, etc. In this paper, we propose a hybrid multi-task learning approach, training on user-item and item-item interactions. We apply item-based contrastive learning on descriptive text, sampling positive and negative pairs based on item metadata. Our approach allows the model to better understand the relationships between entities within the knowledge graph by utilizing semantic information from text. It leads to more accurate, relevant, and diverse user recommendations and a benefit that extends even to cold-start users who have few interactions with items. We perform extensive experiments on two widely used datasets to validate the effectiveness of our approach. Our findings demonstrate that jointly training user-item interactions and item-based signals using synopsis text is highly effective. Furthermore, our results provide evidence that item-based contrastive learning enhances the quality of entity embeddings, as indicated by metrics such as uniformity and alignment.

**Keywords:** knowledge graph neural network, semantic contrastive learning, multi-task learning, cold-start problems, diversity recommendation, content recommendation

## 1. Introduction

The prevalence of content platform services has made a vast collection of media content accessible to consumers. As a result, personalized recommendation systems have improved the user experience by understanding their preference and identifying relevant content. Furthermore, accurate and diverse recommendations have emerged as essential elements within recommendation systems.

Collaborative Filtering (CF) (Miyahara and Paz-zani, 2000; Linden et al., 2003; Hofmann, 2004; Su and Khoshgoftaar, 2006; Jeunen et al., 2022) has been a popular approach in recommendation systems where the system predicts the preferences of a user based on the preferences of similar users. CF-based methods capture user-item interactions by specific operations such as inner products (Sarwar et al., 2001; Wang et al., 2017) or neural networks (He et al., 2017). In the context of this study, we refer to the term ‘item’ as ‘content’ moving forward. However, there are several problems associated with a CF-based model. It requires a significant amount of user-content data in order to prevent data sparsity problems (Srebro et al., 2004; Rennie

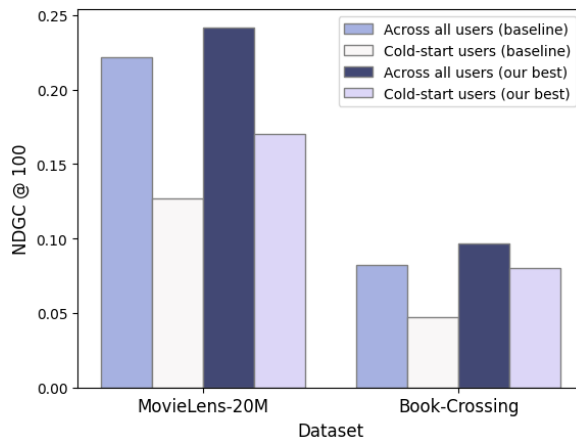


Figure 1: Performance Decline in Cold-Start Scenario. In order to establish a baseline for comparing our model, we take into account the KGCM (Wang et al., 2019b), a pivotal component of KG recommendation models. ‘Cold-start users’ refers to the subgroup of test users positioned in the bottom 1%, signifying those with the most limited user-content interactions.

and Srebro, 2005; Takács et al., 2008). Moreover, it has a cold-start problem (Yu et al., 2004; Adomavicius and Tuzhilin, 2005) which means that it

<sup>†</sup> These authors are corresponding authors.

struggles to make recommendations for new users with no existing data. To tackle these issues, researchers have turned to exploring the addition of user and content features (Cheng et al., 2016) and linking them together (Yu et al., 2014; Zhang et al., 2016; Zhao et al., 2017; Huang et al., 2018; Wang et al., 2018a,b).

Recent research has focused on constructing a Knowledge Graph (KG) that links contents to their attributes and incorporating these graphs into collaborative interactions. Wang et al. (2019b) introduced the Knowledge Graph Convolutional Network (KGCN) as an extension to non-spectral GCN techniques, specifically for recommender systems. Several variant models (Wang et al., 2019a,c, 2020, 2021) utilize new machine learning techniques, such as adding latent variables or network layers. These studies have been proposed with a specific focus on enhancing the performance of rank-based recommendations. While performance improvement is undoubtedly important, overemphasizing ranking-based measures as an evaluation metric can lead to overlooking qualitative factors such as diversity and critical situations such as cold-start. Figure 1 visually demonstrates the decline in performance in the cold-start scenario. Notably, in the domains of both movie and book recommendations, a more significant reduction in the Normalized Discounted Cumulative Gain (NDCG)@100 is evident in the baseline model when compared to our model.

In this paper, we introduce a novel recommendation approach that leverages semantic text information using KGCN to enhance performance from multiple perspectives. Our method uses text embeddings obtained from Pre-trained Language Models (PLMs) to understand the relationships between entities in KG. PLMs (Kenton and Toutanova, 2019; Liu et al., 2019) have addressed numerous limitations of simple one-hot vector encoding. Additionally, the utilization of contrastive learning with PLMs has exhibited superior performance in sentence representation (Radford et al., 2019; Raffel et al., 2019; Oh et al., 2022). However, this approach has not been explored in enhancing the training of KG-based recommendations. Our study investigates the effectiveness of semantic embeddings which are learned through content-based contrastive learning. We apply the proposed method to the MovieLens-20M (Harper and Konstan, 2015) and Book-Crossing<sup>1</sup> datasets. The evaluation of our proposed approach is conducted along three dimensions: (1) Recommendation performance using Click-Through Rate (CTR) such as Area Under the Curve (AUC) and F1-score, along with ranking metrics including Recall and NDCG in both standard and cold-start scenarios, (2) Assessing

the level of personalization and diversity of recommended contents by Inter- and Intra- list diversity metrics, and (3) Measuring the quality of embeddings through metrics that assess uniformity and alignment. The experimental evaluation shows that our proposed method outperforms baseline approaches by taking into account users' personalized interests and diversity in the recommendation process, with an even more substantial performance boost for cold-start users. In addition, the results from the experiments confirm that content-based contrastive learning effectively regularizes content embeddings to achieve better uniformity and alignment of positive pairs in both movie and book domains. Our contributions in this paper are as follows:

- We propose a multi-task learning approach that utilizes user-content and content-content interactions to enrich content recommendations from various perspectives.
- To the best of our knowledge, we are the first to introduce content-based contrastive learning with semantic text for better KG entity relationships, yielding accurate, diverse user recommendations.
- Our empirical assessment affirms the effectiveness of our approach in improving recommendation performance, personalization, diversity, and the quality of content embeddings, extending its utility to demanding scenarios such as those involving cold-start users.
- We provide insights into the importance of comprehensive evaluation and analysis in obtaining valuable recommendations, emphasizing the need to consider multiple factors beyond rank-based performance metrics.

## 2. Related Works

### 2.1. Pre-trained Language Model

Pre-trained Language Models (PLMs) have shown outstanding performance on many Natural Language Processing (NLP) tasks related to understanding the semantic meaning of sentences and context. Fine-tuned Transformer-based (Vaswani et al., 2017) language models such as Bidirectional Encoder Representations from Transformers (BERT) (Kenton and Toutanova, 2019) or Universal sentence encoder (Cer et al., 2018) models have been successfully applied in various domains including text classification (Sun et al., 2019), named entity recognition (Kim et al., 2022), question answering (Yang et al., 2019), and semantic search (Deshmukh and Sethi, 2020; Esteva et al., 2020;

<sup>1</sup><https://grouplens.org/datasets/book-crossing/>

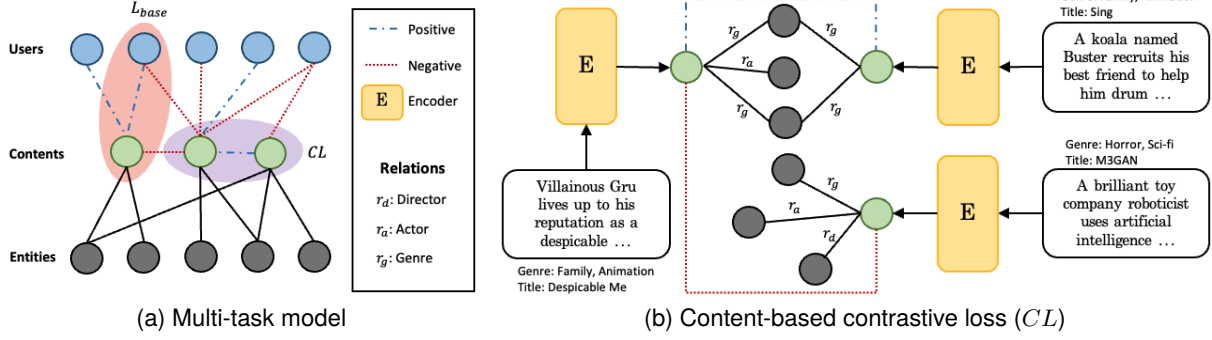


Figure 2: Overview of the Proposed Model. (a) Illustration of a knowledge graph structure and a model with multiple objectives. The loss from user-content interactions is labeled  $L_{base}$ , and the content-based contrastive loss is  $CL$ . (b) Detailed process of our proposed objective function  $CL$ . To generate initial content node embeddings, a pre-trained language model encodes the synopsis of each content. The positive and negative pairs are selected for each content based on their genre or title metadata (as outlined in Section 3.1).

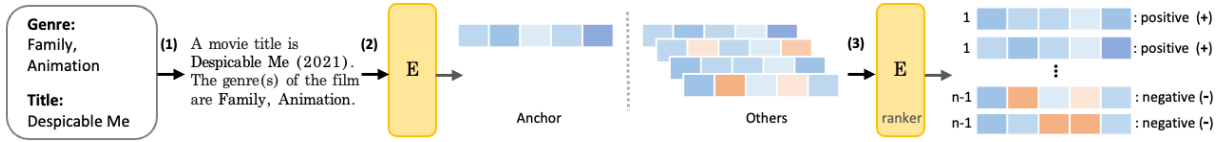


Figure 3: Sampling Positive/Negative Pairs Using Cross-encoder for  $CL$

Rome et al., 2022). These models are capable of capturing more intricate relationships between words and sentences, including contextual information.

**PLMs in Recommendation** Fine-tuned PLMs have gained significant attention in recommendation tasks. Penha and Hauff (2020) worked on the usage of BERT in conversational recommender systems. They highlighted the effectiveness of pre-trained transformer models, particularly BERT, in language modeling and their capacity to implicitly retain factual information for recommendation purposes. Spillo et al. (2023) employed a fine-tuned BERT to encode content-based representations and subsequently integrated them into graph convolutional networks to capture collaborative and structured item data. The pre-trained BERT achieved outstanding breakthroughs in the ranking task as well which is critical in the recommendation task. The BECR model, introduced by Yang et al. (2021), presented a re-ranking scheme that combines deep contextual token interactions with traditional lexical term-matching features. Wu et al. (2021) proposed an end-to-end multi-task learning framework for product ranking with fine-tuned domain-specific BERT to address the issue of vocabulary mismatch between queries and products. Rauf et al. (2024) introduced BCE4ZSR which utilizes fine-tuned transformer models to distill knowledge from a larger encoder to enhance news recommendations.

## 2.2. Contrastive Learning

The objective of Contrastive Learning (CL) is to obtain high-quality embeddings by bringing pairs of samples with similar meanings together in close proximity while pushing apart dissimilar ones (Hadsell et al., 2006). A crucial topic in contrastive learning revolves around the construction of pairs of similar (positive) data points ( $x_i, x_i^+$ ). Obtaining positive pairs is more difficult than dissimilar (negative) pairs because negative examples can be drawn randomly from each training batch. In NLP, researchers devise schemes to augment data sets to produce more positive samples, such as using back-translation, paraphrasing, or generating new sentences using generative models. Similar objectives of contrastive learning have been examined in various contexts (Henderson et al., 2017; Gillick et al., 2019; Karpukhin et al., 2020; Gao et al., 2021). In Equation (1),  $z_i$  represents the embeddings of sentence  $x_i$ ,  $z_i^+$  denotes positive samples that are similar to  $z_i$ . Additionally,  $t$  refers to a temperature hyperparameter suggested by Wu et al. (2018). The loss of  $N$  in-batch negative samples (Chen et al., 2017) can be expressed using the following equation.

$$-\log \frac{e^{(\text{sim}(z_i, z_i^+)/t)}}{\sum_{j=1}^N e^{(\text{sim}(z_i, z_j^+)/t)}} \quad (1)$$

**CL in Recommendation** Park and Lee (2022) proposed contrastive learning for a music recommendation system based on Siamese neural networks (Koch et al., 2015) as follows:

$$L = yD^2 + (1 - y)\max(\text{margin} - D, 0)^2 \quad (2)$$

where  $y$  represents the label assigned to an item pair and  $D$  stands for the distance between the items. When presented with a pair of items labeled as  $y = 1$ , the distance is reduced during training as  $L = D^2$ . Conversely, for a pair of items labeled as  $y = 0$ , the distance gets close to the *margin* value according to  $\max(\text{margin} - D, 0)^2$ . Ma et al. (2023) applied  $K$ -means clustering algorithms to produce prototypes for user and item clusters, aiming to minimize the disparity between these prototypes using a contrastive objective detailed in Equation (1). The work by Li et al. (2023) introduces graph contrastive learnings across multiple and separate domains to explore both domain-shared and domain-specific preference features for target users to improve recommendation performance, especially in sparse interaction scenarios.

### 3. Proposed Methods

Our main goal is to explore how content-based contrastive learning using semantic text affects recommendation performance when it is trained jointly with conventional collaborative loss. In Figure 2 (a), our proposed model is depicted, where we denote the conventional collaborative loss as  $L_{base}$  and our proposed loss as  $CL$ . Further details about the model will be discussed in this section.

#### 3.1. Sampling Strategy

Figure 3 illustrates the positive/negative sampling process to train via  $CL$ . (1) We transform the metadata of the genre or the genre and title into sentences through a simple template. When we use the genre information solely, we create a sentence by utilizing the template: “The genre(s) of the (film|book) is/are [genre\_list].” Likewise, when using the genre and title information together, we follow the template: “A (movie|book) title is (movie\_title|book\_title) (released\_year). The genre(s) of the (film|book) is/are [genre\_list].” (2) The transformed sentences are input to  $E$  and assigned embedding values. (3) The embeddings are passed to the cross-encoder,  $E_{ranker}$ , which takes in a query and the embeddings for processing. The query is one of the embeddings (*Anchor* in Figure 3) and the cross-encoder ranks all the other embeddings according to how similar they are to the query. This process is performed for each embedding. The positive and negative sets are determined by selecting the top- $n$  and bottom  $n$  samples, respectively.

#### 3.2. Content-based Contrastive Loss

To optimize the recommendation model, we adopt conventional user-content interaction loss (Wang et al., 2019b). The assumption is that interactions between the user and the content should be given high scores for positive preferences and low scores for negative preferences. The loss function is as follows:

$$L_{base} = \sum_{u \in U} \left\{ \sum_{c: y_{uc}=1} \mathcal{F}(y_{uc}, \hat{y}_{uc}) - \sum_{i=1}^{S^u} \mathbb{E}_{c_i \sim p(c_i)} \mathcal{F}(y_{uc}, \hat{y}_{uc}) \right\} \quad (3)$$

where  $\hat{y}_{uc}$  is predicted probability,  $\mathcal{F}$  is cross-entropy loss, and  $p$  is a user’s negative preference sampling distribution.  $S^u$  is the number of negative samples for user  $u$ , which is equal to  $|\{c : y_{uc} = 1\}|$ .

Figure 2 (b) illustrates a detailed process of our proposed content-based contrastive loss ( $CL$ ). The loss function is represented by Equation (4).  $C$  is a collection of sentences transformed from metadata of the genre or the genre and title. The positive and negative sets from Section 3.1 are denoted by  $P \subset C$  and  $N \subset C$ , respectively.  $P$  contains samples that are similar to the content while  $N$  contains dissimilar samples.  $h_c$  denotes the representation of the transformed sentences  $c \in C$ . Let  $h_c^+$  and  $h_c^-$  indicate positive and negative sample of  $h_c$ . Inner product,  $\text{sim}(h_c, h_c^+) = h_c \cdot h_c^+$ , is used for our similarity function.

$$CL = \sum_{c \in C} \left\{ \sum_{c^+ \in P} \mathcal{F}(y_{cc^+}, \text{sim}(h_c, h_{c^+}^+)) - \sum_{c^- \in N} \mathcal{F}(y_{cc^-}, \text{sim}(h_c, h_{c^-}^-)) \right\} \quad (4)$$

$$L = \gamma L_{base} + (1 - \gamma)CL + \lambda \|\Theta\|_2^2 \quad (5)$$

Finally, we obtain the loss function to train Equation (3) and (4) jointly in Equation (5), where  $\gamma$  is a balancing hyper-parameter of losses.  $\Theta = \{U, V, R, W_i, b_i, \forall i \in \{0, 1, \alpha\}\}$  is the model parameter set,  $U, V$ , and  $R$  are the embedding tables for all users, nodes (contents and attribute entities), and relations, respectively.  $W_i$  and  $b_i$  denote weights of  $i^{\text{th}}$ -layer and  $\alpha$  layer represents an additional layer specifically dedicated to content embeddings. The last term is an  $L_2$ -regularizer parameterized by  $\lambda$  on  $\Theta$ .

#### 3.3. Training

Our approach follows the training procedure of the Knowledge Graph Convolutional Network (KGCN) (Wang et al., 2019b). KGCN follows the identical node feature update mechanism as spatial GCN. The difference lies in the aggregation of importance scores for each relation across all users. In the equation  $H^{(l+1)} = \sigma(H^{(l)}W^{(l)} + b^{(l)})$ , the variables

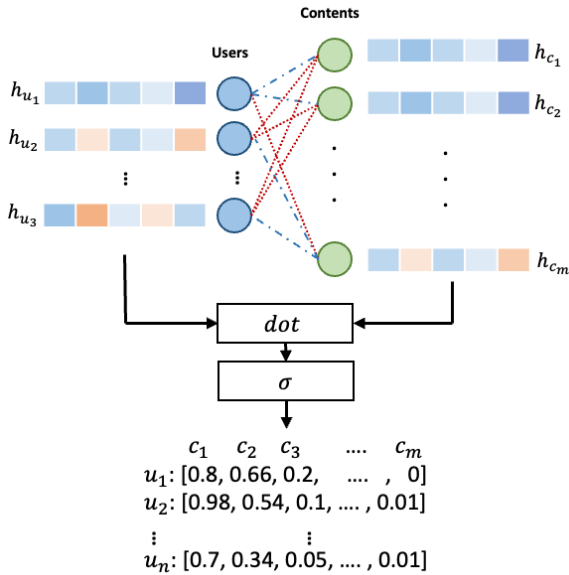


Figure 4: Inference Using the Trained Contents and Users Embedding. The process *dot* signifies the inner product operation, while  $\sigma$  denotes the sigmoid transformation. Each user embedding, indexed by  $n$ , undergoes an inner product operation with each content embedding, indexed by  $m$ , followed by applying the sigmoid transformation. This procedure enables the model to rank all content for each user.

$H^{(l)}$ ,  $W^{(l)}$ , and  $b^{(l)}$  represent the node features, trainable weights, and bias at layer  $l$ , respectively. In a standard GCN,  $H^{(l)}$  is derived through the aggregation, either by summation or concatenation (Hamilton et al., 2017) of a node’s inherent features with those of its neighboring nodes. In contrast, KGCN introduces an additional step where an importance score is applied to each neighbor node’s features before aggregation. KGCN is best optimized through the use of the summation aggregator while we adopt the concatenation aggregator since initializing node features for contents and attribute entities differs in our method. Our content embeddings are derived from the synopsis representation using the encoder  $E$ , while other entity embeddings are initialized randomly. Concatenation ensures that both features are considered when updating the weights (Poursafaei et al., 2022). We represent the initial embedding of contents as  $h_c = MLP(E(x_c))$ , where  $x_c$  corresponds to the synopsis text of each content,  $E$  refers to an encoder and  $MLP$  is a multilayer perception. As the output representation of  $E$  varies in dimensions depending on the encoders used, we utilize an additional  $MLP$  layer to align the embedding dimensions with those of other node embeddings. Note that the  $MLP$  layer only processes content embedding and not other attribute entity embeddings.

### 3.4. Inference

Figure 4 illustrates the sequence of steps in the inference process. After training the graph using KGCN as detailed in Section 3.3, embeddings are generated for users, content, attribute entities, and relations. Notably, attribute entity embeddings and relation embeddings are not utilized in this work. We compute the inner product between each user and all content embeddings, followed by applying a sigmoid transformation. Consequently, we can rank all content items based on individual users, facilitating personalized content recommendations.

## 4. Experiments

In this section, we demonstrate the efficacy of the proposed method under two different scenarios, movie and book recommendations.

### 4.1. Experimental Setup

Table 1: Basic statistics and hyper-parameter settings for the two datasets (u-c: # of user-content interactions, c-c: # of content-content interactions by leveraging genre metadata or both genre and title metadata respectively,  $K$ : neighbor sampling size,  $\eta$ : learning rate,  $d$ : dimension of embeddings,  $\lambda$ :  $L_2$ -regularizer,  $\gamma$ : balancing parameters for losses weight).

	MovieLens-20M	Book-Crossing
# users	138,159	17,860
# contents	16,954	14,967
u-c	13,501,622	139,746
c-c	336,176 / 305,172	221,617 / 223,140
# entities	102,569	77,903
# relations	32	25
# KG triples	499,474	151,500
$K$	4	4
# layers	2	1
optimizer	Adam	Adam
$\eta$	2e-2	2e-4
$d$	32	64
$\lambda$	1e-7	2e-5
$\gamma$	8e-1	8e-1
batch size	65,536 (ours: 30,000)	256
Text source	TMDB <sup>3</sup>	Goodreads <sup>4</sup> , Google Books <sup>5</sup>

**Dataset** We use the MovieLens-20M (Harper and Konstan, 2015) and Book-Crossing<sup>2</sup> datasets. The MovieLens-20M dataset consists of approximately 20 million explicit ratings (ranging from 1 to 5) contributed by users on the MovieLens website. Book-Crossing dataset contains 1 million ratings (ranging from 1 to 10). We follow Wang et al. (2019b) when constructing a Knowledge Graph (KG), utilizing Satori by Microsoft. The basic statistics of the two datasets and KGs are presented in Table 1.

<sup>2</sup><https://grouplens.org/datasets/book-crossing/>

Table 2: Performance of the AUC and F1-score were evaluated and compared against the baseline model. The highest performance is denoted in bold and the second best results are underlined. Statistical significance was assessed using t-tests; significance levels are denoted by  $**p < 0.001$ ,  $*p < 0.05$ .

Dataset	Metric	Without Semantic Text			With Semantic Text		
		$L_{base}$ (Baseline)	$+CL_{genre}$	$+CL_{title+genre}$	$L_{base}$	$+CL_{genre}$	$+CL_{title+genre}$
Movie	<b>AUC</b>	0.9761	0.9773**	0.9775**	0.9763	0.9777**	<b>0.9780**</b>
	<i>p</i> -value	-	0.0007	1.96e-04	0.7290	0.0004	5.33e-05
	<b>F1</b>	0.9290	0.9311*	0.9319**	0.9284	0.9320**	<b>0.9324**</b>
	<i>p</i> -value	-	0.0029	1.21e-04	0.7330	0.0002	6.96e-05
Book	<b>AUC</b>	0.6923	0.6911	0.6946	<b>0.7019**</b>	0.7010**	0.7009**
	<i>p</i> -value	-	0.6217	0.1411	2.39e-05	0.0008	0.0004
	<b>F1</b>	<b>0.6370</b>	0.6336	<u>0.6362</u>	<u>0.6362</u>	0.6360	0.6354
	<i>p</i> -value	-	0.7930	0.1100	0.6011	0.6992	0.5824

We obtain 16,954 movie synopses from the TMDB database<sup>3</sup>. This process entails using the IDs mapped from MovieLens to the corresponding IDs in TMDB which is given by the MovieLens dataset. We attempt to gather synopses for 16,954 contents in our knowledge graph, but 169 of them are missing because they are not available in TMDB. Given the absence of metadata in the Book-Crossing dataset, we leverage ISBN data associated with each book entry to retrieve metadata from Goodreads<sup>4</sup>(Wan and McAuley, 2018; Wan et al., 2019) and Google Books<sup>5</sup>. Out of a total of 14,967 books, this methodology resulted in the successful acquisition of 11,156 genre information and 10,294 synopses.

**Encoders** We utilize two pre-trained language models (Encoder  $E$  in Figure 2 and Figure 3) to represent text and one cross-encoder model (Encoder  $E_{ranker}$  in Figure 3) (Reimers and Gurevych, 2019) for ranking the contents. We use  $BERT_{base}$  (Kenton and Toutanova, 2019) for  $E$  in Figure 2 to embed the initial input of each node’s synopsis.  $E$  in Figure 3 uses *multi-qa-mpnet-base-dot-v1*<sup>6</sup> to encode verbalized metadata.  $E_{ranker}$  uses *ms-marco-MiniLM-L-12-v2*<sup>7</sup> to rank the metadata embeddings according to similarity. We opt for the *multi-qa-mpnet-base-dot-v1* model due to its proven superiority in embedding for semantic search. This model is able to provide optimized metadata embeddings when inputting to the cross-encoder. The cross-encoder model, *ms-marco-MiniLM-L-12-v2* is fine-tuned on MS MARCO Passage Retrieval dataset (Bajaj et al., 2016). This particular model is highly robust in ranking text based

on their similarity, as it has been fine-tuned using a dataset that comprises real user queries and their relevant text from the Bing search engine.

**Hyper-parameters** The hyper-parameters besides batch size remain the same as those used in the original learning process. Unlike the baseline model, which utilizes a batch size of 65,536, we take a batch size of 30,000 for the movie domain. This adjustment is necessary to optimize training for both loss functions due to the significantly lower number of content-content interactions compared to user-content.

## 4.2. Evaluation Metrics

Our evaluation is conducted through Fidelity Jurity<sup>8</sup>, an authoritative research library that provides recommender system evaluations and encompasses three different experimental scenarios. First, we evaluate Click-Through Rate (CTR) prediction by measuring interactions in the test set using the Area Under the Curve (AUC) and F1-score metrics. Second, we use the trained model to select top- $K$  contents with the highest predicted probability for each user in the test set in order to make top- $K$  recommendations. The evaluation is performed through Recall@ $K$  and Normalized Discounted Cumulative Gain (NDCG)@ $K$ . Furthermore, we assess NDCG performance under a cold-start scenario to analyze how it is impacted by varying levels of user activity. Third, we assess the diversity among users’ recommended content lists using the Inter-list diversity@ $K$  metric:

$$\frac{\sum_{i,j,\{u_i,u_j\} \in I} (\cosine\_distance(R_{u_i}, R_{u_j}))}{|I|} \quad (6)$$

where  $U$  denote the set of  $N$  unique users,  $u_i, u_j \in U$  denote the  $i^{th}$  and  $j^{th}$  user in the user set,  $i, j \in \{1, \dots, N\}$ .  $R_{u_i}$  is the binary indicator vector representing provided recommendations for  $u_i$ .  $I$

<sup>3</sup><https://www.themoviedb.org>

<sup>4</sup><https://www.goodreads.com>

<sup>5</sup><https://books.google.com>

<sup>6</sup>[https://www.sbert.net/docs/pretrained\\_models.html](https://www.sbert.net/docs/pretrained_models.html)

<sup>7</sup>[https://www.sbert.net/docs/pretrained\\_cross-encoders.html](https://www.sbert.net/docs/pretrained_cross-encoders.html)

<sup>8</sup><https://github.com/fidelity/jurity>

Table 3: Performance of Recall@K and NDCG@K of the proposed model compared to the baseline model. The highest performance is denoted in bold and the second best results are underlined.

Model	Recall@K					NDCG@K					
	5	10	20	50	100	5	10	20	50	100	
<b>Without Semantic Text</b>											
Movie	$L_{base}$ (Baseline)	0.0734	0.1228	0.1967	0.3214	0.4581	0.0677	0.0970	0.1312	0.1811	0.2216
	+ $CL_{genre}$	0.0659	0.1199	0.1999	0.3337	0.4679	0.0681	0.0972	0.1327	0.1824	0.2244
	+ $CL_{title+genre}$	0.0683	0.1237	0.1976	<u>0.3400</u>	<b>0.4738</b>	0.0706	0.1006	0.1343	0.1866	0.2286
Book	$L_{base}$ (Baseline)	0.0619	0.0741	0.0926	0.1217	0.1585	0.0586	0.0641	0.0690	0.076	0.0825
	+ $CL_{genre}$	0.0636	0.0778	0.0973	0.1317	0.1677	0.0630	0.0688	0.0732	0.0811	0.0873
	+ $CL_{title+genre}$	<b>0.0774</b>	<b>0.0927</b>	<b>0.1115</b>	<u>0.1433</u>	0.1784	0.0596	0.0643	0.0703	0.0793	0.0846
<b>With Semantic Text</b>											
Movie	$L_{base}$ (Baseline)	<u>0.0754</u>	0.1222	0.2016	0.3260	0.4599	0.0753	0.1036	0.1363	0.1834	0.2265
	+ $CL_{genre}$	<b>0.0755</b>	<u>0.1312</u>	<u>0.2082</u>	0.3357	<u>0.4727</u>	<u>0.0810</u>	<u>0.1097</u>	<u>0.1405</u>	<u>0.1911</u>	<u>0.2324</u>
	+ $CL_{title+genre}$	<u>0.0754</u>	<b>0.1325</b>	<b>0.2136</b>	<b>0.3418</b>	0.4714	<b>0.0882</b>	<b>0.1154</b>	<b>0.1498</b>	<b>0.1983</b>	<b>0.2419</b>
Book	$L_{base}$ (Baseline)	0.0425	0.0645	0.0808	0.1245	0.1685	0.0679	0.0742	0.0794	0.0871	0.0938
	+ $CL_{genre}$	<u>0.0730</u>	0.0865	0.1032	<b>0.1468</b>	0.1823	0.0684	0.0756	0.0809	<b>0.0904</b>	<b>0.0968</b>
	+ $CL_{title+genre}$	0.0675	0.0856	<u>0.1041</u>	0.1431	<b>0.1933</b>	<b>0.0744</b>	<b>0.0789</b>	<b>0.0820</b>	<u>0.0892</u>	<u>0.0961</u>

is the set of all unique user pairs,  $\forall i < j, \{u_i, u_j\} \in I$ . Additionally, we calculate the average pairwise cosine distances of the contents recommended to a user based on the content embeddings of our trained model. Then the results from all users are averaged as the metric Intra-list diversity@K:

$$\frac{1}{N} \sum_{i=1}^N \frac{\sum_{p,q,\{p,q\} \in I^{u_i}} (\text{cosine\_distance}(v_p^{u_i}, v_q^{u_i}))}{|I^{u_i}|} \quad (7)$$

where  $v_p^{u_i}, v_q^{u_i}$  are the content features of the  $p^{th}$  and  $q^{th}$  content in the list of contents recommended to  $u_i, p, q \in \{0, 1, \dots, k-1\}$ .  $I^u$  is the set of all unique pairs of content indices for  $u_i, \forall p < q, \{p, q\} \in I^{u_i}$ . Lastly, we verify the quality of our embeddings by using alignment and uniformity metrics (Wang and Isola, 2020). The alignment metric serves to gauge the proximity of features within positive pairs, while the uniformity metric evaluates the distribution of normalized features on the hypersphere.

### 4.3. Main Results

We present the results of our study in tables with abbreviated notation. Our baseline model is referred to as  $L_{base}$  (Baseline), trained identically to the original KGCN model (Wang et al., 2019b). The ‘With Semantic Text’ section of the tables indicates that the respective models were trained using textual input to represent the content embeddings. The notation + $CL_{(genre|title+genre)}$  denotes that a model is trained using content-based contrastive loss and  $L_{base}$  jointly.  $CL_{genre}$  refers to the utilization of content-content interactions extracted from genre metadata, while  $CL_{title+genre}$  indicates the use of such interactions obtained from both genre and title. All results in the tables are averaged over ten random trials.

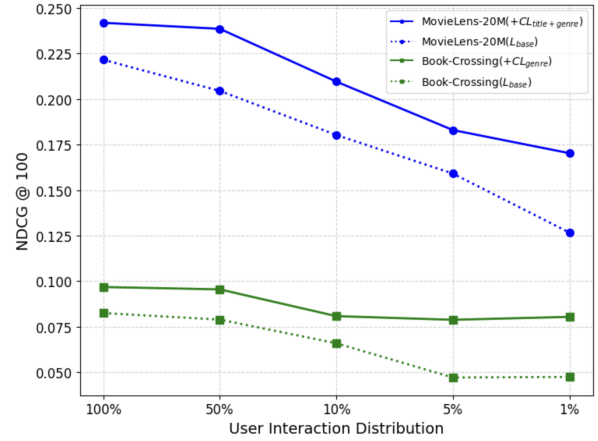


Figure 5: Comparing Model Performance Across a Spectrum of User Activity Levels. The distribution on the x-axis is based on the number of user-content interactions of each test user. For example, 1% refers to the test users with the fewest user-content interactions.

In the CTR prediction experiment, we utilize a t-test, with each model subjected to ten trials and a comparison against the baseline model. Table 2 provides a summary of AUC and F1-score results. The results reveal that the models that employed our content-based contrastive loss function and semantic text information exhibited superior performance overall. The results obtained from Table 3 highlight that our proposed model performs better than the baseline for all  $K$  in the top- $K$  recommendation evaluation. Moreover, Figure 5 underscores that our best model in both domains shows a lesser drop in NDCG@100 for test users with few user-content interactions compared to the baseline.

An evaluation of the diversity of recommendations in our models is performed with the find-

Table 4: Performance of Intra- and Inter- List diversity of the proposed model compared to the baseline model. The highest performance is denoted in bold and the second best results are underlined.

Dataset	Metric	Without Semantic Text			With Semantic Text		
		$L_{base}$ (Baseline)	$+CL_{genre}$	$+CL_{title+genre}$	$L_{base}$	$+CL_{genre}$	$+CL_{title+genre}$
Movie	Inter@20	0.7054	0.7478	0.7511	0.6952	<u>0.7615</u>	<b>0.7806</b>
	Intra@20	0.1997	0.3427	0.2667	0.1970	<b>0.4407</b>	<u>0.4201</u>
Book	Inter@20	0.2963	0.3207	0.3196	0.3374	<u>0.3494</u>	<b>0.3670</b>
	Intra@20	0.0048	0.0029	0.0024	<b>0.0168</b>	0.0135	<u>0.0162</u>

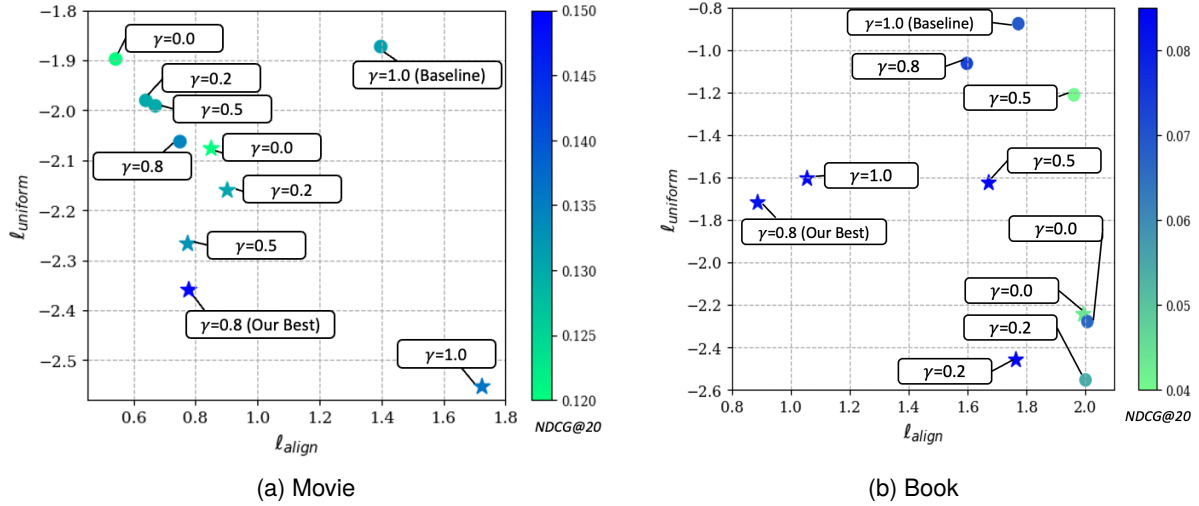


Figure 6: Performance of Embedding Quality.  $l_{align}$ - $l_{uniform}$  plot of models by  $\gamma$  which represents a weight of base loss,  $L_{base}$ . Embeddings obtained from the trained model using synopsis data are marked with star points( $\star$ ), while circle points( $\bullet$ ) indicate embeddings obtained from the model trained without synopsis data. The color of points represents the NDCG@20. For both  $l_{align}$  and  $l_{uniform}$ , lower numbers are better.

ings presented in Table 4. Our proposed model, trained using  $CL_{genre+title}$ , achieved the highest performance in Inter-list diversity, while  $CL_{genre}$  or  $L_{base}$  (with semantic text) exhibited the best performance in Intra-list diversity. In the context of CTR or ranking-based performance, we found that in most cases, the use of both genre and title resulted in superior performance. However, such an approach may not always be optimal in terms of diversity. For instance, in the movie domain, if we assume that the user rates Despicable Me highly, then Despicable Me 2 might also be recommended due to a similar title. Consequently, it can boost ranking-based performance when a user consumes the recommended content. However, due to their similar embeddings, it adversely impacts the diversity performance of the recommendations. For this reason, we conclude that using  $CL_{genre+title}$  scored lower diversity performance compared to  $CL_{genre}$ . It is imperative to clarify the Intra-list diversity result of the book domain, as they exhibit significantly lower numbers in comparison to others. This is due to the fact that each user typically engages

in approximately seven user-content interactions, in contrast to the movie dataset, where there are approximately 100 interactions. Furthermore, the low Intra-list Diversity scores imply that the recommended items for each user exhibit significant similarities and this trend becomes more conspicuous when the model is trained without semantic text. This finding contributes to the examination of the subsequent embedding quality result (uniformity and alignment). Figure 6 presents a comparison of the uniformity and alignment of embeddings generated by various models trained using the  $CL$  loss function, based on different  $\gamma$  values, along with their corresponding NDCG@20 scores. In general, models that have superior alignment and uniformity demonstrate better performance (Wang and Isola, 2020). In the movie domain, we observed that the utilization of semantic text mostly produces better uniformity in models, regardless of the value of  $\gamma$ . Additionally, it can be observed that as the value of gamma decreases, the alignment value also tends to decrease. If a model does not use the semantic text information, it can be inferred that



Table 5: Comparison of performance utilizing text generated by a generative AI, LLaMA to the model using human-generated text and the baseline model. Recall@20 and NDCG@20 are respectively represented by R@20 and N@20.

Dataset	Model	R@20	N@20	Inter	Intra
Movie	$L_{base}$ (Baseline)	0.1967	0.1312	0.7054	0.1997
	+ $CL_{title+genre}$ (Human-generated text)	0.2136	0.1498	0.7806	0.4201
	+ $CL_{title+genre}$ (LLaMA-generated text)	0.2106	0.1504	0.7717	0.4268
Book	$L_{base}$ (Baseline)	0.0926	0.0690	0.2963	0.0048
	+ $CL_{title+genre}$ (Human-generated text)	0.1041	0.0820	0.3670	0.0162
	+ $CL_{title+genre}$ (LLaMA-generated text)	0.1092	0.0807	0.3437	0.0151

the embeddings are highly anisotropic (Ethayarajh, 2019) since the alignment value decreases significantly while the uniformity value does not decrease much. However, this tendency is less prominent in the book domain, aligning with our previous observations made during the examination of Intra-list diversity. This can be explained by the relatively low average number of user-content interactions and the significant similarity in item embeddings. When the user-content loss balancing parameter,  $\gamma$  is set to a low value, the training process faces challenges in achieving effective convergence, as illustrated by the data points in the lower right corner of Figure 6 (b). Thus, by calibrating the balancing parameter  $\gamma$  in accordance with the specific dataset, we can acquire well-aligned embeddings with enhanced diversity in recommendation performance. Our best model, trained with loss functions  $L_{base}$  and  $CL$  with a  $\gamma$  value of 0.8, demonstrates robust alignment and uniformity in the experimental results.

#### 4.4. Ablation Studies

In Section 4.1, we observed that a total of 169 synopses were missing from the movie dataset, 4,673 were absent from the book dataset, and the model training proceeded without utilizing them. It is uncertain whether this omission would have any significant impact on the overall performance of the model. In addition, it is worth noting that manual annotation of data is often a resource-intensive and time-consuming process. As such, we seek to explore whether the use of generative AI models could potentially aid in mitigating this issue. We generated 16,954 movie and 14,967 book synopses by providing a clear and concise prompt, "Briefly, a story of a (movie|book) named (movie\_title|book\_title) released (released\_year) is about " to LLaMA<sup>9</sup> (Touvron et al., 2023). As detailed in Table 5, the results indicate that the model trained using LLaMA-generated text delivers comparable performance with that of the model utiliz-

ing human-generated text, sourced from TMDb, Goodreads, and Google books with regard to diversity and ranking performance.

## 5. Conclusion and Future Work

This paper proposes a hybrid recommendation approach that uses semantic text and a Knowledge Graph (KG) in a multi-task learning framework. We propose a novel content-based contrastive loss function that is jointly optimized with a conventional collaborative loss function. The content-based contrastive loss function utilizes a sampling strategy that creates positive and negative sets for each content based on its metadata. In this study, we aim to explore the efficacy of multi-task learning on KG neural networks that incorporate semantic text information for content recommendations. We conduct our experiments to assess the performance of the recommendation system from various perspectives. The results of our study demonstrate that the proposed model surpasses the baseline in terms of performance metrics based on ranking, diversity, and embedding quality as well as when confronted with a cold-start scenario. Through our work, we provide insights into the importance of comprehensive evaluation and analysis in obtaining valuable recommendations, emphasizing the need to consider multiple factors beyond rank-based performance metrics.

Furthermore, our methods are agnostic to the choice of model. By utilizing our proposed sampling strategy and content-based contrastive objective, any KG-based model can be employed. We performed intensive experiments on Knowledge Graph Convolutional Networks (KG-CNN) to demonstrate the effects of semantic embeddings and a content-based multi-task learning approach in content recommendation. Our future research will focus on two possible avenues: (1) extending the proposed methods to other KG-based neural network models, and (2) assessing the suitability of our approach within the context of real-world recommendation data.

<sup>9</sup><https://ai.facebook.com/blog/large-language-model-llama-meta-ai>

## Ethical Considerations

Since our models were trained using the TMDB, Goodreads, and Google Books dataset or the generated text of the LLaMA, generative language model, there is a risk that they may propagate any toxic or hateful content, such as racism, insults, or xenophobia, present in the training data. To mitigate this, we suggest content moderation in preprocessing to ensure our recommendations adhere to ethical standards.

## Acknowledgements

This work was supported in part by National Science Foundation grant 212720.

## 6. Bibliographical References

- Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174.
- Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 767–776.
- Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishikesh Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10.
- Anup Anand Deshmukh and Udhav Sethi. 2020. Irbert: leveraging bert for semantic search in background linking for news articles. *arXiv preprint arXiv:2007.12603*.
- Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wenpeng Yin, Dragomir Radev, and Richard Socher. 2020. Co-search: Covid-19 information retrieval with semantic search, question answering, and abstractive summarization. *arXiv preprint arXiv:2006.09595*.
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.
- Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. *arXiv preprint arXiv:1909.10506*.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.
- Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.
- F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.
- Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.
- Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.
- Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115.

- Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 505–514.
- Olivier Jeunen, Jan Van Balen, and Bart Goethals. 2022. Embarrassingly shallow auto-encoders for dynamic collaborative filtering. *User Modeling and User-Adapted Interaction*, 32(4):509–541.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.
- Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- Juae Kim, Yejin Kim, and Sangwoo Kang. 2022. Weakly labeled data augmentation for social media named entity recognition. *Expert Systems with Applications*, 209:118217.
- Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.
- Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access*, pages 193907–193934.
- Yakun Li, Lei Hou, Dongmei Li, and Juanzi Li. 2023. Hkgcl: Hierarchical graph contrastive learning for multi-domain recommendation over knowledge graph. *Expert Systems with Applications*, 233:120963.
- Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yubin Ma, Xuan Zhang, Chen Gao, Yahui Tang, Linyu Li, Rui Zhu, and Chunlin Yin. 2023. Enhancing recommendations with contrastive learning from collaborative knowledge graph. *Neuro-computing*, 523:103–115.
- Koji Miyahara and Michael J Pazzani. 2000. Collaborative filtering with the simple bayesian classifier. In *Pacific Rim International conference on artificial intelligence*, pages 679–689. Springer.
- Dongsuk Oh, Yejin Kim, Hodong Lee, H Howie Huang, and Heuseok Lim. 2022. Don't judge a language model by its last layer: Contrastive learning with layer-wise attention pooling. *arXiv preprint arXiv:2209.05972*.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Minju Park and Kyogu Lee. 2022. Exploiting negative preference in content-based music recommendation with contrastive learning. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 229–236.
- Roberto Pellegrini, Wenjie Zhao, and Iain Murray. 2022. Don't recommend the obvious: estimate probability ratios. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 188–197.
- Gustavo Penha and Claudia Hauff. 2020. What does bert know about books, movies and music? probing bert for conversational recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 388–397.
- Farimah Poursafaei, Zeljko Zilic, and Reihaneh Rabbany. 2022. A strong node classification baseline for temporal graphs. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 648–656. SIAM.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Muhammad Arslan Rauf, Mian Muhammad Yasir Khalil, Weidong Wang, Qingxian Wang, Muhammad Ahmad Nawaz Ul Ghani, and Junaid Hassan. 2024. Bce4zsr: Bi-encoder empowered by teacher cross-encoder for zero-shot cold-start news recommendation. *Information Processing Management*, 61(3):103686.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

- Jasson DM Rennie and Nathan Srebro. 2005. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719.
- Scott Rome, Sardar Hamidian, Richard Walsh, Kevin Foley, and Ferhan Ture. 2022. Learning to rank instant search results with multiple indices: A case study in search aggregation for entertainment. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3412–3416.
- Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.
- Giuseppe Spillo, Cataldo Musto, Marco Polignano, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2023. Combining graph neural networks and sentence encoders for knowledge-aware recommendations. In *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, pages 1–12.
- Nathan Srebro, Jason Rennie, and Tommi Jaakkola. 2004. Maximum-margin matrix factorization. *Advances in neural information processing systems*, 17.
- Xiaoyuan Su and Taghi M Khoshgoftaar. 2006. Collaborative filtering for multi-class data using belief nets algorithms. In *2006 18th IEEE international conference on Tools with Artificial Intelligence (ICTAI'06)*, pages 497–504. IEEE.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.
- Gábor Takács, István Pilászy, Botyán Németh, and Domonkos Tikk. 2008. Investigation of various matrix factorization methods for large recommender systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 1–8.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Lyle H Ungar and Dean P Foster. 1998. Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, volume 1, pages 114–129. Menlo Park, CA.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM conference on recommender systems*, pages 86–94.
- Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. *arXiv preprint arXiv:1905.13416*.
- Hongwei Wang, Jia Wang, Miao Zhao, Jiannong Cao, and Minyi Guo. 2017. Joint topic-semantic-aware social recommendation for online voting. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 347–356.
- Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018a. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 417–426.
- Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018b. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 1835–1844.
- Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019a. 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*, pages 968–977.
- Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019b. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*, pages 3307–3313.
- Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

- Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019c. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958.
- Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference 2021*, pages 878–887.
- Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. 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*, pages 219–228.
- Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1652–1656.
- Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*.
- Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742.
- Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*.
- Yingrui Yang, Yifan Qiao, Jinjin Shao, Mayuresh Anand, Xifeng Yan, and Tao Yang. 2021. Composite re-ranking for efficient document search with bert. *arXiv preprint arXiv:2103.06499*.
- Kai Yu, Anton Schwaighofer, Volker Tresp, Xiaowei Xu, and H-P Kriegel. 2004. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):56–69.
- Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norrick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 283–292.
- Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362.
- Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101.
- Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 635–644.