Collaborative Alignment for Recommendation

Chen Wang
University of Illinois Chicago
Chicago, Illinois, USA
cwang266@uic.edu

Liangwei Yang
University of Illinois Chicago
Chicago, Illinois, USA
lyang84@uic.edu

Zhiwei Liu
Salesforce AI Research
Palo Alto, USA
zhiweiliu@salesforce.com

Xiaolong Liu
Mingdai Yang
University of Illinois Chicago
Chicago, Illinois, USA
xliu262@uic.edu
myang72@uic.edu

Yueqing Liang
Illinois Institute of Technology
Chicago, Illinois, USA
yliang40@hawk.iit.edu

Philip S. Yu
University of Illinois Chicago
Chicago, Illinois, USA
psyu@uic.edu

Abstract

Traditional recommender systems have relied primarily on identity representations (IDs) to model users and items. Recently, the integration of pre-trained language models (PLMs) has enhanced the capability to capture semantic descriptions of items. However, while PLMs excel in few-shot, zero-shot, and unified modeling scenarios, they often overlook the crucial signals from collaborative filtering (CF), resulting in suboptimal performance when sufficient training data is available. To effectively combine semantic representations with the CF signal and improve the performance of the recommender system in warm and cold settings, two major challenges must be addressed: **(1) bridging the gap between semantic and collaborative representation spaces**, and (2) **refinement while preserving the integrity of semantic representations**. In this paper, we introduce CARec, a novel model that adeptly integrates collaborative filtering signals with semantic representations, ensuring alignment within the semantic space while maintaining essential semantics. We present experimental results from four real-world datasets, which demonstrate significant improvements. Using collaborative alignment, CARec also shows remarkable effectiveness in cold-start scenarios, achieving notable improvements in recommendation performance. The code is available at https://github.com/ChenMetanoia/CARec.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Recommender System, Collaborative Filtering, Pre-trained Language Model

ACM Reference Format:
Chen Wang, Liangwei Yang, Zhiwei Liu, Xiaolong Liu, Mingdai Yang, Yueqing Liang, and Philip S. Yu. 2024. Collaborative Alignment for Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679535

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679535

1 Introduction

In the contemporary digital landscape, recommendation systems (RecSys) have emerged as indispensable tools, greatly enhancing user experiences across a wide range of online platforms, including e-commerce, content streaming, and social media networks [1]. These systems are instrumental in guiding users towards content, products, or services that align with their preferences, thus significantly contributing to user satisfaction on online platforms. Traditionally, modern recommendation models have relied on unique identifiers (IDs) to represent both users and items, transforming these IDs into embedding vectors through learnable parameters to effectively capture and predict user preferences [7, 23, 27, 35, 36].

Despite the notable success of ID-based recommendation systems (IDRec) in scenarios where sufficient user-item interaction data is available, commonly referred to as the warm setting, their dependency on historical interactions poses significant limitations. Specifically, IDRec struggle to generate reliable recommendations in situations characterized by sparse or non-existent user-item interactions, known as the cold start problem [2, 6, 37, 38]. To mitigate these challenges, Semantic-Based Recommendation Models (SemRec) [12] have been introduced that take advantage of textual content enrich the recommendation process with additional context and information about users and items.

Recent breakthroughs in pre-trained language models (PLMs) have significantly advanced the representation of textual information in the field of Natural Language Processing (NLP)[4, 22, 25]. These advancements provide a valuable opportunity to enhance the efficacy of semantic recommendation systems (SemRec). A pivotal question arises: Can we effectively merge identifier-based recommendation systems (IDRec) with SemRec to leverage the strengths of both, thereby improving recommendations in both warm and cold settings? Attempts to integrate IDRec with SemRec through methods such as addition or concatenation have not yielded the expected improvements[43]. This shortfall may stem from the fundamental differences in their representation spaces: IDRec, which learns from user-item interactions, functions as a collaborative signal, whereas SemRec, derived from text via semantic encoders, serves as a semantic signal. Despite the significant enhancements

in semantic embeddings provided by modern encoders, these still fall short of the performance achieved by IDRec, especially in warm settings where SemRec underperforms [43]. This issue could be attributed to the fact that semantic embeddings in SemRec are often pre-trained on tasks not directly related to recommendation [4, 22]. While they capture meaningful content, they fail to fully represent the specific preferences and distribution patterns of the recommendation data. This scenario presents two distinct challenges. Firstly, **bridging the gap between collaborative and semantic representation spaces is crucial for leveraging the combined strengths of IDRec and SemRec**. Secondly, **refining and maintaining the accuracy of semantic representations during alignment remains a key obstacle**. The "tightness" of semantic representations, as previously discussed [5, 10], can impair model performance when item representations are overly similar. Additionally, altering the semantic representation space risks compromising the meaningfulness of these representations.

To address the previously discussed challenges, we introduce a novel training strategy named CARec (Collaborative Alignment for Recommendation). This approach innovatively integrates IDRec with SemRec in a unique manner. Unlike traditional methods that merge identifier and semantic information through addition or concatenation [43], our framework assigns distinct roles to users and items. Specifically, users are represented by ID embeddings, which are randomly initialized due to the typical absence of direct textual information. Conversely, items are represented by semantic embeddings, initialized through Pre-trained Language Models (PLMs) using textual data such as titles, features, and descriptions. During training, we design two sequential phases to systematically address the aforementioned challenges. The first phase, termed the semantic aligning phase, aims to bridge the gap between collaborative and semantic representation spaces. Unlike the traditional simultaneous training of user and item embeddings, which can contaminate the semantic integrity of item representations when combined with the randomly initialized user IDs. We align ID representation space into semantic representation space. The second phase, known as the collaborative refining phase, focuses on addressing the issue of semantic representation "tightness" while preserving the meaningfulness of these representations.

In summary, our contributions are outlined as follows:

(1) Novel Alignment Paradigm: We introduce a novel collaborative learning paradigm which enhances the quality of recommendations by fostering dynamic knowledge exchange.
(2) Bridging Gaps: We identify the problem, uncover the challenges, and propose a feasible solution for collaborative alignment to bridge the gap between collaborative filtering and semantic representation.
(3) Empirical Validation: We conduct extensive experiments on four real-world datasets under both warm and cold settings to validate the effectiveness of CARec.

2 Problem Definition

In this section, we introduce a new framework termed "Collaborative Alignment". Collaborative Alignment seeks to fuse semantic information with collaborative filtering to provide more accurate personalized recommendations. It is an inevitable problem that occurs between pre-trained language models and collaborative filtering-based recommendations.

For recommendation task, we have a set of users $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$, a set of items $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$ and a historical interaction matrix $\mathbf{R}$ of size $|\mathcal{U}| \times |\mathcal{I}|$. By treating $\mathbf{R}$ as the adjacent matrix, we can also view the historical interaction as a user-item bipartite graph $\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E}) = \{(u, i)|u \in \mathcal{U}, i \in \mathcal{I}, (u, i) \in \mathcal{E}\}$, where $\mathcal{E}$ is the edge set. There is an edge $(u, i) \in \mathcal{E}$ if $\mathbf{R}_{u,i} = 1$ with implicit feedback. In collaborative alignment, besides the collaborative filtering signal $\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E})$, we also have a semantic embedding for each user/item with rich semantic information represented by $\mathbf{x}_u$ and $\mathbf{x}_i$, respectively. semantic embedding is encoded from context information with corresponding pre-trained models as illustrated in Section 3.1. Encoded from pre-trained models, semantic embedding contains rich semantic information, and collaborative alignment seeks to bridge the semantic embedding with the collaborative filtering signal to provide more accurate recommendation.

3 Proposed Method

In this section, we introduce our innovative recommendation model CARec, which is designed to address the challenge of enhancing RecSys by effectively incorporating semantic information and Collaborative Filtering (CF) signals. Using historical user-item interactions, CARec learns comprehensive semantic CF-incorporated representations for both users and items. These representations not only successfully merge semantic and CF signals but also yield substantial improvements in recommendation performance, benefiting both general recommendation scenarios and challenging item cold-start scenarios. In the following subsections, we detail the architecture, training phase, and inference phase of CARec, providing a comprehensive overview of our recommendation approach.

3.1 Semantic Item Representation

To leverage the semantic-rich encoding capabilities offered by pre-trained language models (PLMs), our approach allows for using any PLM as the encoder to capture semantic item embeddings. For a given item i with associated semantic features, including the item title, category, and brand, we concatenate these features into a single sentence $S_i = [w_1, w_2, ..., w_c]$, where w is the text token and c is the total token number. S_i is then used as input to the PLM, resulting in the following semantic representation for item i:

$$\mathbf{x}_i = PLM(S_i), \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^{d_W}$ is i's semantic embedding, and d_W denotes the PLM's output embedding size. We freeze $\mathbf{x}_i$ as the 0-th layer hidden representation $\mathbf{h}_i^{(0)}$ for graph convolution in Section 3.2. This representation captures the rich semantic information from the item's semantic attributes, laying the foundation for the fusion of semantic and collaborative filtering signals within CARec.

3.2 Graph Aggregator

CARec is built upon graph aggregation to spread the rich semantic semantic embedding obtained from Section 3.1. With the aggregation over $\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E})$, CARec updates user/item embedding based on the collaborative filtering signal. Let's denote the embeddings of users as $\mathbf{h}_u$ and the embeddings of items as $\mathbf{h}_i$, which is obtained

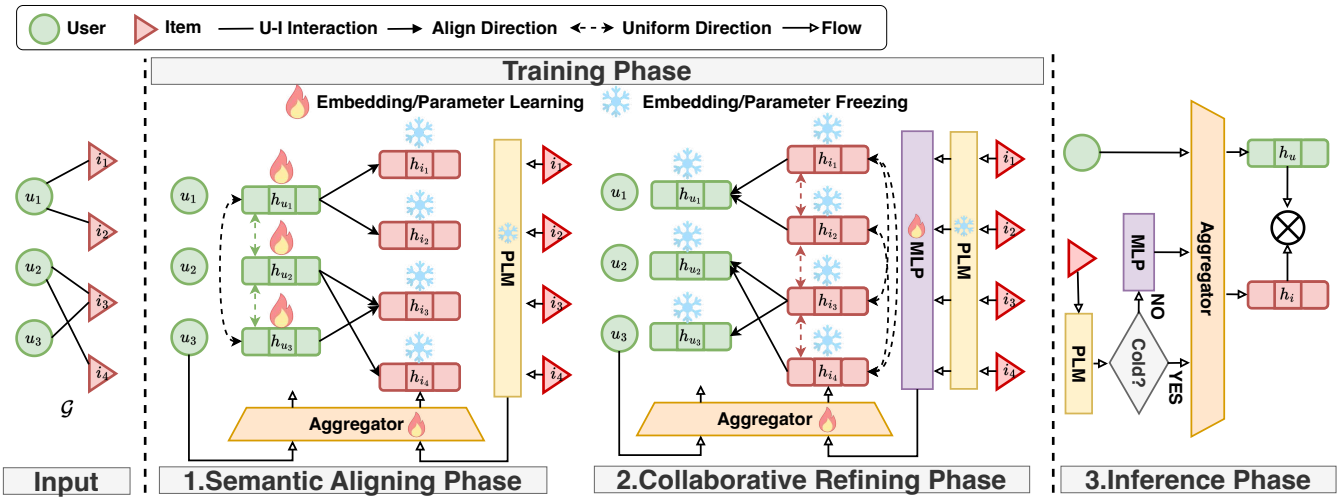


Figure 1: CARec comprises three key phases: the semantic aligning phase, the collaborative refining phase, and inference phase. During the semantic aligning phase, the model aligns user representations with the item semantic representation space. In contrast, the collaborative refining phase focuses on guiding item representations to effectively incorporate collaborative signals while preserving their semantic characteristics. Finally, in the inference Phase, the model leverages the acquired knowledge to provide personalized recommendations by utilizing the learned user embeddings and transformed item embeddings.

by the graph aggregator:

$$\mathbf{h}_u, \mathbf{h}_i = \text{Aggregator}(\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E}), \mathbf{h}_u^{(0)}, \mathbf{h}_i^{(0)}), \tag{2}$$

where $\mathbf{h}_u^{(0)}$ and $\mathbf{h}_i^{(0)}$ represent the user/item initial embedding. $\mathbf{h}_i^{(0)}$ is encoded from Section 3.1 and $\mathbf{h}_u^{(0)}$ is randomly initialized embedding due to the lack of sufficient context to encode semantic embedding. The Aggregator performs aggregation on $\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E})$ for K layers to smooth the embedding. For each layer's aggregation, the computation is defined as:

$$\mathbf{h}_u^{(k+1)} = \mathbf{h}_u^{(k)} + \text{AGG}\left(\mathbf{h}_i^{(k)}, \forall i \in \mathcal{N}(u)\right), \tag{3}$$

$$\mathbf{h}_i^{(k+1)} = \mathbf{h}_i^{(k)} + \text{AGG}\left(\mathbf{h}_u^{(k)}, \forall u \in \mathcal{N}(i)\right), \tag{4}$$

where $\mathbf{h}_u^{(k)}$ represents the embedding of user u at layer k, and $\mathbf{h}_i^{(k)}$ represents the embedding of item i at layer k. The function AGG denotes the aggregation function, which combines the embeddings of neighboring nodes. To maintain generality, we use the most widely used LGCN [7] as the aggregation function:

$$\text{AGG}(\mathbf{h}_i, \forall i \in \mathcal{N}(u)) = \sum_{i \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(u)|}\sqrt{|\mathcal{N}(i)|}} \mathbf{h}_i, \tag{5}$$

where $\mathcal{N}(u)$ and $\mathcal{N}(i)$ represent the set of neighboring nodes of u and i. $\mathbf{h}_i$ is the embedding of a item node i. User aggregation is computed in the same way. It's worth noting that the aggregation function can be replaced with any graph aggregation function. In the subsequent section, we dive into the CARec training phase, where we elucidate the process of learning comprehensive semantic representations incorporated in CF for users and items, a crucial step in our innovative recommendation model.

3.3 Training Phase

In traditional bipartite graph learning for Recommender Systems (RecSys), it is common to initialize user and item embeddings based on identifiers (IDs) randomly and to update their representations symmetrically. This approach involves aggregating neighbors' representations for each user and item.

However, integrating semantic information into this symmetric learning framework often proves suboptimal, primarily due to significant discrepancies in initialization methods. Users are typically represented through ID-based random embeddings, whereas items utilize text-based embeddings generated by pre-trained language models (PLMs). This asymmetric foundation can lead to notable challenges, such as "item representation contamination," where the aggregation of randomly initialized user IDs may dilute the richer, text-based item representations during the process of forming a unified representation space. To overcome these challenges, CARec introduces a novel approach through its semantic aligning phase. This phase stabilizes the item's semantic representation while aligning the user's collaborative signals within the semantic representation space, specifically addressing the representation gap and contamination issues. Subsequently, the collaborative refining phase maintains the aligned user collaborative signal and refines the item's semantic representation within the semantic space. This step is crucial for preserving the integrity and meaningfulness of the semantic content.

Further details on the semantic aligning phase and the collaborative refining phase will be provided in the following subsections, highlighting how each contributes to overcoming traditional limitations in symmetric learning frameworks.

3.3.1 Semantic Aligning Phase. In this phase, we align ID representation space into semantic representation space to address the representation gap challenge.

Leveraging the rich semantic information contained in item semantic representations, our objective is to align user embeddings ($\mathbf{h}_u$) into the item representation space($\mathbf{h}_i$) which has been shown in the Fig. 1. To achieve this alignment and strengthen the normalized element-wise similarity between a user's representation and those of their interacted items, we employ an alignment loss inspired by the DirectAU [28] method. The alignment loss between user u and item i is defined as:

$$l^{\mathcal{U}}_{align} = \frac{1}{|\mathcal{E}|} \sum_{(u,i)\in\mathcal{E}} ||\mathbf{h}_u - \text{freeze}(\mathbf{h}_i)||^2, \tag{6}$$

where freeze($\mathbf{h}_i$) indicates the frozen item embedding.

To prevent over-concentration in the representation space, the uniformity loss is also added as the regularization. In our context, we compute and apply the user uniformity loss $l^{\mathcal{U}}_{uniform}$ to optimize the learning of user representations efficiently:

$$l^{\mathcal{U}}_{uniform} = log\frac{1}{|\mathcal{U}|^2} \sum_{u\in\mathcal{U}} \sum_{u*\in\mathcal{U}} e^{-2||\mathbf{h}_u - \mathbf{h}_{u*}||}. \tag{7}$$

These two loss metrics work in synergy to maintain proximities between positive instances while dispersing random instances across the hypersphere. The final loss function in the user representation learning stage is a combination of the alignment loss and the user uniformity loss:

$$\mathcal{L}_{\mathcal{U}} = l^{\mathcal{U}}_{align} + l^{\mathcal{U}}_{uniform}. \tag{8}$$

The phase of user representation learning concludes upon meeting the convergence criteria, which are predicated on the prediction scores achieved on the validation set. Specifically, we employ NDCG@10 as the benchmark metric, with an early stopping parameter set at 30. At this point, we presume that users have assimilated adequate knowledge from both semantic and collaborative filtering signals, given the current state of item semantic representations. Following this, the subsequent collaborative refining phase is dedicated to refining these item semantic representations.

3.3.2 Collaborative Refining Phase. Contrast to aligning user representations with item semantic representations to integrate collaborative and semantic signals, the collaborative refining phase introduces subtle adjustments when refining item representations. The primary goal in this phase is to preserve item embeddings within the semantic representation space, thus retaining semantic information while incorporating collaborative signals. To achieve this, we freeze user representations and learn from well-trained users. Rather than directly fine-tune item representations, we employ an adaptor, such as multilayer perceptron (MLP) to transform them, as illustrated in Fig. 1 Collaborative Refining Phase. There are two main reasons for this approach. First, item semantic representations alone often fail to capture collaborative filtering signal, and they tend to become densely clustered [5, 9, 10], which can hinder recommendation performance. Second, this approach allows us to preserve informative item semantic knowledge for use in the cold setting directly. Using an MLP to adapt new item representations in the cold setting can be problematic. The primary issue is that the process involves mapping item representations before aggregation. In the warm setting, the MLP learns to adjust the representations for effective aggregation. However, in the cold setting, where new items lack prior interactions, the MLP, being a global learner trained on warm data, struggles to appropriately adjust the item semantic representations in the absence of aggregation data.

To refine item representations, we initially apply MLP, resulting in $\widetilde{\mathbf{h}}^{(0)}_i = \text{MLP}(\mathbf{h}^{(0)}_i)$, where $\widetilde{\mathbf{h}}^{(0)}_i$ denotes the transformed item representations. Subsequently, we compute the convoluted user and item representations as follows:

$$\mathbf{h}_u, \widetilde{\mathbf{h}}_i = \text{Aggregator}(\mathcal{G}(\mathcal{U}, \mathcal{I}, \mathcal{E}), \mathbf{h}^{(0)}_u, \widetilde{\mathbf{h}}^{(0)}_i). \tag{9}$$

The alignment loss employed in item representation learning mirrors the one used in user representation learning but utilizes the transformed item representation $\widetilde{\mathbf{h}}_i$. The alignment loss in the collaborative refining phase is formulated as follows:

$$l^{\mathcal{I}}_{align} = \frac{1}{|\mathcal{E}|} \sum_{(u,i)\in\mathcal{E}} ||\text{freeze}(\mathbf{h}_u) - \widetilde{\mathbf{h}}_i||^2, \tag{10}$$

where freeze($\mathbf{h}_u$) is the frozen user embedding to force the training on the item. In addition to the alignment loss, we introduce a uniformity loss for items to prevent over-concentration and ensure a well-distributed representation space. This uniformity loss encourages item representations to maintain suitable distances from each other, thereby promoting diversity in the recommendation process. The uniformity loss for items is defined as follows:

$$l^{\mathcal{I}}_{uniform} = \log \frac{1}{|\mathcal{I}|^2} \sum_{i\in\mathcal{I}} \sum_{i*\in\mathcal{I}} e^{-2||\widetilde{\mathbf{h}}_i - \widetilde{\mathbf{h}}_{i*}||}. \tag{11}$$

It fosters the even distribution of item representations within the hypersphere, thereby enhancing the model's ability to capture nuanced differences between items with similar semantic features.

The final loss function for item representation learning is a combination of the alignment loss and the item uniformity loss:

$$\mathcal{L}_{\mathcal{I}} = l^{\mathcal{I}}_{align} + l^{\mathcal{I}}_{uniform}. \tag{12}$$

Through this approach, we ensure that item representations capture both semantic information and collaborative filtering signals, leading to improved recommendation quality while retaining the flexibility to address cold-start problems.

3.4 Inference Phase

During the inference phase, CARec leverages the learned user embeddings and transformed item embeddings to make personalized recommendations. The recommendation score $s(u, i)$ for a user-item pair (u, i) is determined through the dot product between the user embedding $\mathbf{h}_u$ and the transformed item embedding $\widetilde{\mathbf{h}}_i$:

$$s(u, i) = \mathbf{h}^T_u \cdot \widetilde{\mathbf{h}}_i. \tag{13}$$

In the cold setting, we using the dot product between the user embedding $\mathbf{h}_u$ and the item semantic embedding $\mathbf{h}_i$ to calculate the recommendation score.

Table 1: The Statistics of Preprocessed Datasets: "Avg.U" represents the average number of interactions per user, "Avg.I" signifies the average number of interactions per item, and "Cold-Items" indicates the count of newly introduced items.

	Electronic	Office Products	Gourmet Food	Yelp
#Users	81,512	51,493	66,268	65,870
#Items	32,424	16,920	24,636	43,215
#Inters	623,896	212,795	307,617	831,470
#Avg.U	7.653	4.133	4.642	12.623
#Avg.I	18.232	11.950	11.857	19.240
#Cold-Items	1,797	888	1,310	1,714

4 Experiments

This section empirically evaluates the proposed CARec on four real-world datasets. The goal is to answer the four following research questions (RQs). **RQ1:** What is the performance of CARec? **RQ2:** Does CARec still achieve the best in the challenging cold-start recommendation? **RQ3:** How do different parts affect CARec? **RQ4:** Can CARec really keep the rich semantic semanticized information?

4.1 Experimental Setup

4.1.1 Dataset. To rigorously evaluate the performance of our proposed methodology, we conduct experiments in both warm and cold settings. Key statistics of the preprocessed datasets are summarized in Table 1. Specifically, we use four publicly available real-world datasets from the Amazon Review Dataset[1]: Electronics, Office Products, and Grocery and Gourmet Food and Yelp Dataset[2]. These datasets have been widely employed in prior recommendation system studies [9, 10].

4.1.2 Baselines. We compare the proposed approach with the following baseline methods: **NeuMF** [8] is a neural network-enhanced matrix factorization model that replaces the conventional dot product with a multi-layer perceptron (MLP) to capture more nuanced user-item interactions. **DirectAU** [28] introduces an innovative loss function that evaluates representation quality in collaborative filtering (CF) based on alignment and uniformity within the hypersphere. In our implementation, we employ the alignment and uniformity loss, updating only the student role. **NGCF** [30] aggregates information from neighbor- ing nodes and incorporates collaborative signals into embeddings. **PinSage** [42] designed a random walk strategy for large-scale graphs, specifically on the Pinterest platform. **LightGCN** [7] represents a state-of-the-art recommendation algorithm grounded on Graph Convolutional Networks (GCN) [14]. It enhances performance by omitting feature transformations and nonlinear activations. **SimpleX** [20] proposes an easy-to-understand model with a unique loss function that incorporates a larger set of negative samples and employs a threshold to eliminate less informative ones. It also utilizes relative weights to balance the contributions of positive-sample and negative-sample losses. **NCL** [18] offers a neighborhood-enriched contrastive learning framework tailored for graph collaborative filtering. It explicitly captures both structural and semantic neighbors as objects for contrastive learning. **Wide&Deep** [3] is a context-aware recommendation model that trains both wide linear models and deep neural networks concurrently, aiming to synergize the benefits of both memorization and generalization in RecSys. **DCNV2** [29] is another context-aware recommendation model that enhances the expressive power of Deep & Cross Networks (DCN) by extending the original weight vector into a matrix. **DroupoutNet** [26] employs a dropout operation during training, randomly discarding portions of the collaborative embeddings. **Heater** [46] utilizes the sum squared error (SSE) loss to model collaborative embeddings based on content information.

4.1.3 Evaluation Settings. We evaluate our model's recommendation performance using commonly employed metrics in the field of Recommender Systems (RecSys): Recall@K and NDCG@K. By default, we set the values of K to 10 and 50. The reported results are based on the average scores across all users in the test set. These metrics consider the rankings of items that users have not interacted with yet. In line with established practices [7, 8], we utilize a full-ranking technique, which involves ranking all non-interacted items for each user. To assess the model's performance in a cold setting, we follow the procedures outlined in previous studies [2, 26, 31, 46]. In the Cold-start scenario, we identify cold-start items by removing all training interactions for randomly selected subsets of items.

To ensure the validity of both warm and cold settings, we apply meticulous preprocessing to these datasets. Initially, in alignment with previous work [10], we employ a 5-core filtering strategy, eliminating users and items with insufficient interactions. Subsequently, we randomly select 5% of items to serve as the cold-start items, excising all corresponding interactions from the preprocessed dataset to create a cold-start item dataset. This ensures that cold-start items are only encountered during the testing phase. The remaining dataset is partitioned into training, validation, and testing subsets using an 80%-10%-10% split. For the semantic features of items, we aggregate information from fields such as *title*, *categories*, and *brand* in the Amazon dataset, truncating any item text exceeding 512 tokens. To prepare the Yelp dataset for analysis, we started by removing any items that did not have textual information. We then focused on interactions with ratings of 3 or higher. From the filtered set, we designated 5% of the items as "cold items" to create a specialized dataset for evaluating cold-start performance. We further narrowed down the dataset to include only those items and users involved in at least 15 interactions each, ensuring a more focused and relevant dataset for analysis. The remaining data was then split randomly: 80% was used for training the model, and the remaining 20% was equally divided between validation and testing purposes. The Yelp dataset, with its combination of business IDs (representing items) and rich textual descriptions (such as categories), provides an excellent opportunity to evaluate our model across a wide range of scenarios.

4.1.4 Implementation Details. We implement CARec and other baseline models using the open-source recommendation library, RecBole[3] [34]. For the sake of a fair comparison, we employ the Adam optimizer across all methods and conduct meticulous hyperparameter tuning. The batch size is configured at 1,024, and

[1] https://jmcauley.ucsd.edu/data/amazon/
[2] https://www.yelp.com/dataset
[3] https://recbole.io/docs/index.html

we implement early stopping with a patience setting of 30 epochs to mitigate overfitting, using NDCG@10 as our evaluation metric. Item text embeddings are generated utilizing a pretrained *Instructor-xl* model[4] [25]. The dimensions for both user and item embeddings are fixed at 768.

For residual hyperparameters, we employ a grid search strategy to identify optimal settings. Specifically, the learning rate is explored within the set {0.0001, 0.001, 0.01}, and the weight decay coefficient is tuned among $\{1e^{-4}, 1e^{-5}, 1e^{-6}\}$. For graph-based models, the number of layers is evaluated over {1, 2, 3}. For content-based models, item semantic representations serve as item features. Within the MLP in CARec, we explore configurations with layer counts in {1, 2, 3}, hidden dimensions in {384, 768, 1536}, and dropout rates in {0.2, 0.5}.

4.2 RQ1: Evaluation of the recommendations

We evaluate the performance of our proposed method against various baseline approaches across four distinct datasets, with results detailed in Table 2. For the sake of clarity and comparative analysis, we categorize the baseline methods into three distinct classes: ID-based (denoted as $_{\text{ID}}$), Text-based (denoted as $_{\text{TEXT}}$), and content-based, which includes models such as Wide&Deep and DCNV2. In ID-based models, the embedding for both the user and item is learned exclusively from their respective IDs and interactions. Text-based models, on the other hand, utilize semantic representations of items to initialize item embeddings. These semantic representations are obtained from the *instructor-xl* [25] model. Additionally, for the Text-based baseline, we compute user representations by averaging the embeddings of items they have interacted with, as generated by *instructor-xl*, and then use a dot product with item semantic representations for making recommendations. For content-based models, on the other hand, incorporate item semantic representation as features for enhanced semantic understanding.

In a comparative analysis with established baseline methods, our proposed CARec model consistently demonstrates superior performance over both ID-based and Text enhanced algorithms across a wide range of evaluation metrics. It validates that CARec can effectively incorporate collaborative filtering signal and semantic information to set the state-of-the-art performance.

4.3 RQ2: Cold-start Evaluation

Table 3 provides an overview of the results of cold-start item recommendations. Key findings include: 1) Our model, CARec, consistently outperforms the best-performing baseline models in cold-start item recommendations. In contrast, DropoutNet and Heater, despite utilizing the same item semantic representations as item features, and employing pre-trained Matrix Factorization (MF) to initialize user and item representations, fall short in comparison to our model. This underscores the efficacy of CARec in leveraging item semantic knowledge and highlights the benefits of its unified representational space, seamlessly integrated into the item semantic representation space. This integration serves a dual purpose: it retains the semantic richness of item semantic descriptions while effectively addressing the cold-start item challenge. Importantly, CARec accomplishes this without the need for auxiliary modules or additional steps. 2) collaborative-based models struggle to harness the informative item semantic representation for addressing the cold-start item problem. This challenge arises from the Semantic Disparity between semantic and Collaborative spaces. Treating users and items as equal entities during training leads to an alignment of the unified representation space into a new space that diverges from the original item semantic representation space. Consequently, when a new item is introduced, its semantic representation deviates from the unified representation space, causing the model to encounter difficulties in making recommendations for the new item. 3) The performance ranking of baseline models shows notable variations in the Yelp dataset. Specifically, SimpleX$_{\text{TEXT}}$ outperforms DropoutNet, while NCL$_{\text{TEXT}}$ experiences a marked decline. This shift can be attributed to the nature of text data in the Yelp dataset, which generally consists of shorter and simpler textual content compared to the comprehensive product descriptions found in the Amazon dataset.

[4]https://huggingface.co/hkunlp/instructor-xl

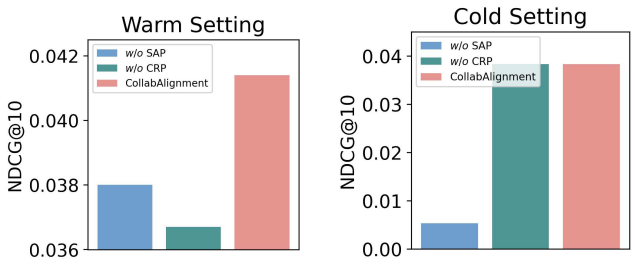


Figure 2: Ablation study of CARec on Electronic

4.4 RQ3: In-depth Analysis

4.4.1 Ablation Study. In this subsection, we present a comprehensive analysis of the impact of each proposed technique and component on both the warm and cold setting performances. To facilitate a thorough comparison, we prepare two variants of the CARec model: (1) a variant without semantic aligning phase (SAP), denoted as w/o SAP, maintaining the training strategy consistent with collaborative models; and (2) a variant omits the collaborative refining phase, applying only item semantic representation, denoted as w/o CRP.

The results of this ablation study are illustrated in Fig. 2. Notably, the absence of SAP, as observed in w/o SAP, leads to a significant reduction in model performance. This underscores the critical importance of preserving the item semantic representation space. Additionally, the exclusion of the collaborative refining phase (CRP) reveals interesting insights. In the warm setting, omitting the CRP harms performance, as item semantic representations alone may tend to crowd together. Conversely, in the cold setting, item semantic representations still retain valuable context features that assist in addressing the cold-start challenge.

4.4.2 Impact of Diverse Pretrained Language Models (PLMs). Numerous robust Pretrained Language Models (PLMs) hold the potential to enhance RecSys by providing valuable item semantic representations. To identify the most effective PLMs for our specific objective, we conducted an evaluation of item semantic representations

Table 2: Warm Setting Comparison Table. The best and second-best results are bold and underlined, respectively. $_{\text{ID}}$ indicates ID-based models, and $_{\text{TEXT}}$ denotes models that employ item semantic representation for item embedding initialization. "*" denotes that the improvements are significant at the level of 0.05 with paired t-test.

	Electronic				Office Products				Grocery and Gourmet Food				Yelp			
Model	R@10	R@50	N@10	N@50	R@10	R@50	N@10	N@50	R@10	R@50	N@10	N@50	R@10	R@50	N@10	N@50
NeuMF_{ID}	0.0513	0.0675	0.0298	0.0339	0.1599	0.1868	0.1051	0.1113	0.1390	0.1644	0.0892	0.0951	0.0469	0.1326	0.0277	0.0498
$\text{DirectAU}_{\text{ID}}$	0.0546	0.0694	0.0292	0.0329	0.1661	0.1965	0.1000	0.1069	0.1438	0.1707	0.0844	0.0907	0.0467	0.1440	0.0277	0.0529
NGCF_{ID}	0.0431	0.0598	0.0232	0.0299	0.0935	0.1248	0.0674	0.0745	0.0854	0.1146	0.0604	0.0672	0.0392	0.1298	0.0231	0.0478
$\text{PinSage}_{\text{ID}}$	0.0447	0.0613	0.0249	0.0321	0.0956	0.1273	0.0694	0.0765	0.0897	0.1174	0.0640	0.0704	0.0413	0.1332	0.0258	0.0484
$\text{LightGCN}_{\text{ID}}$	0.0530	0.0857	0.0331	0.0413	0.1782	0.2177	0.1217	0.1307	0.1526	0.1917	0.1024	0.1114	0.0498	0.1441	0.0291	0.0535
$\text{SimpleX}_{\text{ID}}$	0.0558	0.1060	0.0305	0.0429	0.1727	0.2172	0.1091	0.1192	0.1519	0.1989	0.0920	0.1028	0.0458	0.1355	0.0277	0.0508
NCL_{ID}	0.0558	0.1032	0.0348	0.0473	0.1831	0.2316	0.1267	0.1361	0.1573	0.2063	0.1046	0.1175	0.0531	0.1566	0.0309	0.0578
Instructor-xl	0.0108	0.0255	0.0061	0.0094	0.0028	0.0121	0.0012	0.0031	0.0016	0.0045	0.0012	0.0018	0.0008	0.0029	0.0004	0.0010
$\text{NeuMF}_{\text{TEXT}}$	0.0389	0.0540	0.0237	0.0275	0.1271	0.1579	0.0846	0.0916	0.1166	0.1417	0.0736	0.0795	0.0255	0.0835	0.0151	0.0301
$\text{DirectAU}_{\text{TEXT}}$	0.0551	0.0718	0.0294	0.0336	0.1671	0.1963	0.1004	0.1071	0.1434	0.1703	0.0843	0.0906	0.0472	0.1430	0.0278	0.0526
$\text{NGCF}_{\text{TEXT}}$	0.0137	0.0477	0.0064	0.0136	0.0619	0.2145	0.0785	0.0973	0.1061	0.1929	0.0652	0.0845	0.0416	0.0824	0.0267	0.0358
$\text{PinSage}_{\text{TEXT}}$	0.0114	0.0452	0.0051	0.0123	0.0902	0.1221	0.0636	0.0708	0.0867	0.1121	0.0609	0.0667	0.0342	0.0870	0.0188	0.0307
$\text{LightGCN}_{\text{TEXT}}$	0.0560	0.0909	0.0320	0.0407	0.1782	0.2177	0.1217	0.1307	0.1525	0.1965	0.1030	0.1131	0.0496	0.1433	0.0290	0.0533
$\text{SimpleX}_{\text{TEXT}}$	0.0514	0.0931	0.0334	0.0437	0.1759	0.2121	0.1202	0.1284	0.1513	0.1891	0.1015	0.1102	0.0495	0.1405	0.0294	0.0531
NCL_{TEXT}	0.0553	0.0929	0.0324	0.0417	0.1767	0.2213	0.1174	0.1275	0.1547	0.2074	0.1021	0.1142	0.0068	0.0264	0.0042	0.0093
Wide&Deep	0.0138	0.0485	0.0074	0.0159	0.1014	0.1428	0.0609	0.0705	0.0923	0.1303	0.0554	0.0642	0.0219	0.0756	0.0128	0.0266
DCNV2	0.0373	0.0575	0.0216	0.0266	0.1292	0.1648	0.0829	0.0910	0.1160	0.1516	0.0717	0.0799	0.0248	0.0837	0.0145	0.0296
CARec	**0.0641***	**0.1073**	**0.0414***	**0.0520***	**0.1880***	**0.2317**	**0.1348***	**0.1389***	**0.1634***	**0.2103***	**0.1174***	**0.1281***	**0.0582***	**0.1731***	**0.0329***	**0.0622***

Table 3: Cold Setting Comparison Table. Notations consistent with the warm setting comparison. "*" denotes that the improvements are significant at the level of 0.05 with paired t-test.

	Electronic				Office Products				Grocery and Gourmet Food				Yelp			
Model	R@10	R@50	N@10	N@50	R@10	R@50	N@10	N@50	R@10	R@50	N@10	N@50	R@10	R@50	N@10	N@50
Instructor-xl	0.0488	0.1197	0.0297	0.0456	0.0374	0.1401	0.0159	0.0382	0.0091	0.0595	0.0053	0.0158	0.0067	0.0343	0.0037	0.0108
$\text{NeuMF}_{\text{TEXT}}$	0.0046	0.0250	0.0021	0.0066	0.0101	0.0558	0.0047	0.0145	0.0074	0.0413	0.0034	0.0107	0.0030	0.0220	0.0014	0.0058
$\text{DirectAU}_{\text{TEXT}}$	0.0048	0.0277	0.0023	0.0073	0.0118	0.0511	0.0057	0.0143	0.0082	0.0429	0.0038	0.0113	0.0049	0.0229	0.0026	0.0068
$\text{NGCF}_{\text{TEXT}}$	0.0015	0.0175	0.0021	0.0058	0.0095	0.0513	0.0050	0.0138	0.0072	0.0351	0.0033	0.0092	0.0033	0.0217	0.0032	0.0057
$\text{PinSage}_{\text{TEXT}}$	0.0032	0.0194	0.0023	0.0063	0.0114	0.0528	0.0051	0.0141	0.0091	0.0360	0.0042	0.0100	0.0038	0.0235	0.0041	0.0063
$\text{LightGCN}_{\text{TEXT}}$	0.0104	0.0443	0.0049	0.0124	0.0309	0.0960	0.0173	0.0315	0.0119	0.0522	0.0053	0.0140	0.0047	0.0201	0.0023	0.0059
$\text{SimpleX}_{\text{TEXT}}$	0.0098	0.0419	0.0049	0.0120	0.0122	0.0606	0.0064	0.0168	0.0113	0.0583	0.0056	0.0158	0.0260	0.0801	0.0144	0.0272
NCL_{TEXT}	0.0188	0.0716	0.0092	0.0209	0.0338	0.1150	0.0143	0.0320	0.0222	0.0778	0.0128	0.0248	0.0051	0.0275	0.0023	0.0076
Wide&Deep	0.0038	0.0204	0.0018	0.0055	0.0140	0.1118	0.0060	0.0267	0.0150	0.1224	0.0072	0.0297	0.0042	0.0260	0.0021	0.0071
DCNV2	0.0057	0.0282	0.0029	0.0078	0.0107	0.0510	0.0047	0.0134	0.0082	0.0356	0.0039	0.0099	0.0045	0.0265	0.0023	0.0074
DroupoutNet	0.0569	0.1211	0.0349	0.0503	0.1317	0.1931	0.0850	0.0985	0.1481	0.2432	0.0984	0.1212	0.0236	0.0781	0.0120	0.0236
Heater	0.0036	0.0293	0.0017	0.0073	0.0078	0.0412	0.0034	0.0105	0.0271	0.0500	0.0095	0.0146	0.0052	0.0269	0.0041	0.0074
CARec	**0.0622***	**0.1501***	**0.0383***	**0.0584***	**0.1469***	**0.2646***	**0.0931***	**0.1210***	**0.1764***	**0.3012***	**0.1219***	**0.1521***	**0.0283***	**0.0851***	**0.0163***	**0.0292***

generated by four additional PLMs. These PLMs have achieved varying rankings on the Massive Text Embedding Benchmark (MTEB) leaderboard [5], highlighting their diverse capabilities and potential contributions to the enhancement of RecSys. The five PLMs evaluated include *instructor-xl*[25], *all-MiniLM-L6-v2*[22], *all-mpnet-base-v2*[22], and *bge-base-en-v1.5*[33] and *bert-base-unchased*[4]. The experimental results are presented in Table 4, demonstrating significant improvements in the cold setting than the default model. This underscores the vital importance of aligning the user representation space with the item semantic representation space. Notably, *instructor-xl* emerges as the top-performing PLM overall, as it can generate text embeddings simply by providing the task instruction, without requiring fine-tuning. For our experiments, we used the instruction "Represent the Amazon title:" with *instructor-xl* to generate the text embeddings.

4.4.3 Should We Include Additional Item Tutoring and collaborative refining phases? To investigate the advantages of further training user and item representations beyond the initial item tutoring and collaborative refining phases, we conducted experiments involving continuous learning on these representations. Fig. 3 presents the model's performance in both warm and cold settings across three datasets. In the plot, "Item Tut" indicates that the current phase is the semantic aligning phase, while "User Tut" designates the collaborative refining phase. The numbers on the x-axis represent the current training stage.

Table 4: Comparison Table of PLMs. Notations consistent with warm setting comparison.

Electronic	Warm Setting		Cold Setting	
PLMs	R@10	N@10	R@10	N@10
instructor-xl	0.0641	0.0414	**0.0622**	**0.0383**
all-MiniLM-L6-v2	0.0633	0.0414	0.0563	0.0343
all-mpnet-base-v2	**0.0641**	**0.0419**	0.0556	0.0341
bge-base-en-v1.5	0.0636	0.0414	0.0561	0.0339
bert-base-unchased	0.0631	0.0411	0.0352	0.0194

[5] https://huggingface.co/spaces/mteb/leaderboard

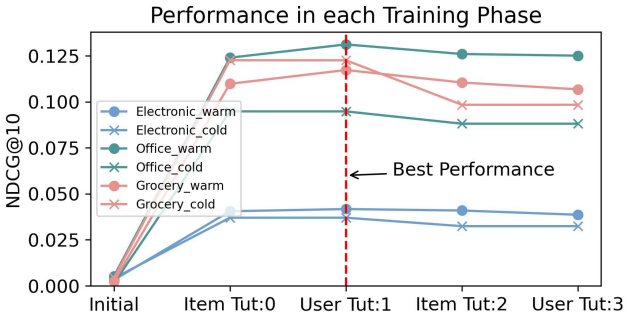


Figure 3: Overall performance in each training phase

Table 5: Impact of User Embedding Initialization on Model Performance in the Yelp Dataset.

	Warm				Cold			
	R@10	R@50	N@10	N@50	R@10	R@50	N@10	N@50
CARec	0.0578	0.1720	0.0332	0.0627	**0.0278**	**0.0842**	**0.0155**	**0.0288**
$\text{CARec}_{\text{AVG}}$	**0.0579**	**0.1722**	**0.0342**	**0.0637**	0.0221	0.0735	0.0119	0.0240

As depicted in Fig. 3, CARec achieved its best performance after the first collaborative refining phase, denoted as "User Tut:1," across all three datasets. This suggests that user and item representations do not require additional separate training stages. Continuing to train the model beyond this point results in a performance decline, possibly due to the significant deviation of user and item representations from the item semantic representation, leading to a loss of semantic information.

4.4.4 Should users be initialized with semantic representations instead of random initialization? In the Yelp dataset, as illustrated in Table 5, we explore the effect of initializing user embeddings through average pooling of historical item sequences, denoted as $\text{CARec}_{\text{AVG}}$. Our findings are as follows: (1) Utilizing average pooling for initializing user embeddings enhances model performance in scenarios with abundant historical data (warm setting) but results in diminished effectiveness in data-scarce situations (cold setting). This indicates that while average pooling can somewhat narrow the representation gap, it does not offer the same level of adaptability as random initialization, especially in contexts with sparse user-item interactions. (2) Although average pooling helps bridge the initial representation gap, it does not reach the full potential of performance enhancement unless coupled with a robust training framework like ours. (3) The comparative analysis of $\text{CARec}_{\text{AVG}}$ and CARec underscores that the choice of embedding initialization strategy can significantly influence outcomes, contingent upon its integration within a systematic training methodology.

4.4.5 Parameter Sensitivity. We experimented with different configurations of the number of layers for the MLP. The sensitivity results, shown in Fig. 4, reveal that the layer number is not sensitive for model performance, and the highest performance is achieved when using two layers with a hidden dimension of 768.

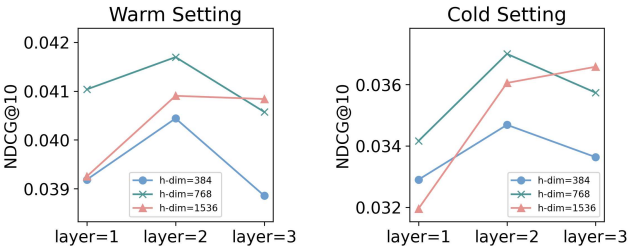
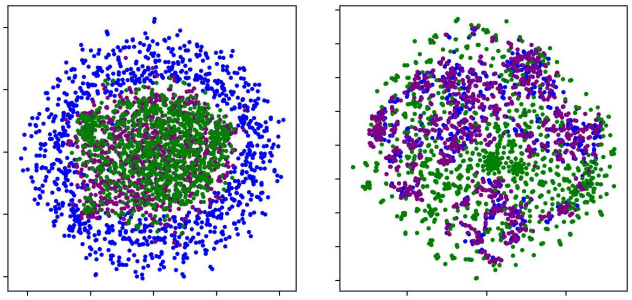


Figure 4: Parameter analysis of MLP on Electronic



(a) CF Representation Space **(b) CA Representation Space**

Figure 5: Comparison of representation space after model alignment. The left figure illustrates the representation space following Collaborative Filtering (CF) Alignment, while the right figure depicts the representation space after Collaborative Alignment (CA). In both figures, the blue nodes symbolize item semantic representations, the purple nodes represent item mapped representations by MLP, and the green nodes denote user learned representations. CA ensures that item semantic representations remain in the same space after MLP transformation.

4.5 RQ4: Case Study

To provide visual evidence of CARec's effectiveness in aligning user representations with item semantic representations while preserving the integrity of item-learned representations, we present a case study in Fig 5. In the left figure, which represents the representation space following traditional collaborative filtering alignment, the item semantic representation (blue) is shown surrounding the user (green) and item (purple) mapped representations by MLP. This spatial arrangement suggests that the item semantic representation is not effectively integrated into the same space as the user and item representations. In contrast, CARec, as shown in the right figure, successfully aligns the user and item learned representations within the item semantic representation space, ensuring that informative semantic information is retained. This alignment contributes to significant improvements in both warm and cold settings, showcasing the model's enhanced performance.

5 Related work

5.1 Aligning CF with Semantic Representations

Several studies have attempted to align collaborative filtering (CF) signals with semantic representations in recommender systems. KAR [32] utilizes Large Language Models (LLMs) to enhance recommendation systems by incorporating open-world knowledge and reasoning capabilities about user preferences and item information. This approach marks a significant advancement in integrating real-world knowledge into recommendation systems. Similarly, CTRL [17] explores the integration of collaborative and semantic signals for Click-Through Rate (CTR) prediction, demonstrating how semantic insights from Pre-trained Language Models (PLMs) can be combined with collaborative data to improve recommendation accuracy. LC-Rec [44] introduces an innovative method of semantic integration using tree-structured vector quantization within LLMs, enhancing how different semantic representations interact within recommendation contexts. This model emphasizes the evolution of semantic integration technologies and their application in recommender systems. CoWPiRec [41] focuses on integrating collaborative filtering information into text-based item representations through a novel word graph that captures word-level collaborative signals. This technique enhances PLM's by incorporating user interaction data directly, offering improvements in cross-domain and cold-start recommendation scenarios. Unlike these existing works, our approach proposes a unique model that not only integrates but also refines and aligns these representations more effectively. We focus on dynamically adjusting both user and item embeddings to address the limitations of static embedding approaches commonly seen in the current literature, specifically targeting the gaps in representation alignment and the optimization of semantic integrity.

5.2 Collaborative Filtering

Collaborative Filtering (CF) is a widely used technique in modern RecSys. CF models typically represent users and items as embeddings and learn these embeddings by reconstructing historical user-item interactions. With the rise of Graph Neural Networks (GNNs) [14], GNN-based RecSys have gained popularity. These methods model user-item interactions as bipartite graphs, enabling them to capture high-order connectivity. SpectralCF [45] introduced spectral convolution to improve recommendation performance, particularly for cold-start items. PinSAGE [42] designed a random walk strategy for large-scale graphs, specifically on the Pinterest platform. NGCF [30] aggregates information from neighboring nodes and incorporates collaborative signals into embeddings. LightGCN [7] simplified NGCF, achieving better performance and reduced training time. However, these collaborative models treat users and items equally and learn their representations simultaneously, which may not work well when incorporating informative item semantic representations.

5.3 Cold-start Recommendation

Addressing the cold-start problem requires bridging the gap between warm-start and cold-start items. To achieve this, side information, particularly content features, is often integrated into CF-based recommendation models. These content features serve as a link to capture the collaborative signal for cold-start items. For example, models like DropoutNET [26] and CC-CC [24] randomly omit certain collaborative embeddings, enhancing the robustness of CF-based models while implicitly tapping into information related to the collaborative signal from item content features. In contrast, some approaches focus on explicitly modeling the correlation between content information and collaborative embeddings [19, 40, 46]. In our approach, we take a different path by directly combining collaborative filtering signals with content information through collaborative alignment. This innovative approach significantly enhances recommendation performance by seamlessly blending collaborative and content-based information.

5.4 Pre-trained Language Models

General text embeddings are of paramount importance, finding wide utility not only in common applications such as web search and question answering[13] but also in their foundational role in enhancing large language models [11, 15]. Unlike task-specific methods, general text embeddings must be versatile and applicable across various contexts. In recent years, significant strides have been made in this field, resulting in notable works like BERT [4], Instructor [25], sentence-T5 (Ni et al., 2021a), Sentence-Transformer [22], C-Pack [33], OpenAI text embedding [21], and more. In the realm of RecSys, existing research has demonstrated the power and effectiveness of Pre-trained Language Models (PLMs) in enhancing RecSys [5, 9, 10, 16, 39], particularly in warm, cold, and few-shot settings. However, most of these studies have primarily focused on sequential recommendation scenarios, while our work centers on collaborative filtering-based recommendations.

6 Conclusion

In this paper, we study the collaborative alignment problem to bridge the gap between collaborative filtering and the pre-trained language model. Taking advantage of the pre-trained language model, we first obtain the item semanticized embedding with rich semantic information. Then, we propose CARec to encode user embedding into the item's semantic embedding space based on the collaborative signal. CARec treats users and items in different roles to better utilize both the collaborative filtering signal and the rich semantic information on items. Experiments on three real-world datasets under both warm and cold settings show our proposed CARec surpasses current state-of-the-art methods. Our case study on learned embedding space highlights that CARec can keep the semantic information on the semantic embedding space from pre-trained language model.

Acknowledgements

This work is supported in part by NSF under grants III-2106758, and POSE-2346158.

References

[1] Ashton Anderson, Lucas Maystre, Ian Anderson, Rishabh Mehrotra, and Mounia Lalmas. 2020. Algorithmic effects on the diversity of consumption on spotify. In *WWW 2020*. 2155–2165.

[2] Yuwei Cao, Liangwei Yang, Chen Wang, Zhiwei Liu, Hao Peng, Chenyu You, and Philip S. Yu. 2023. Multi-task Item-attribute Graph Pre-training for Strict Cold-start Item Recommendation. In *Recsys 2023*. ACM, 322–333. https://doi.org/10.1145/3604915.3608806

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *RecSys 2016*. ACM, 7–10. https://doi.org/10.1145/2988450.2988454

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[5] Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. 2021. Zero-Shot Recommender Systems. *CoRR* abs/2105.08318 (2021). arXiv:2105.08318 https://arxiv.org/abs/2105.08318

[6] Ziwei Fan, Zhiwei Liu, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. 2022. Sequential recommendation via stochastic self-attention. In *WWW 2022*. 2036–2047.

[7] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR 2020*. 639–648.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW 2017*. ACM, 173–182. https://doi.org/10.1145/3038912.3052569

[9] Yupeng Hou, Zhankui He, Julian J. McAuley, and Wayne Xin Zhao. 2022. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. *CoRR* abs/2210.12316 (2022). https://doi.org/10.48550/arXiv.2210.12316 arXiv:2210.12316

[10] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. [n. d.]. Towards Universal Sequence Representation Learning for Recommender Systems. In *KDD 2022*.

[11] Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot Learning with Retrieval Augmented Language Models. *CoRR* abs/2208.03299 (2022). https://doi.org/10.48550/arXiv.2208.03299 arXiv:2208.03299

[12] Umair Javed, Kamran Shaukat, Ibrahim A Hameed, Farhat Iqbal, Talha Mahboob Alam, and Suhuai Luo. 2021. A review of content-based and context-based recommendation systems. *iJET 2021* 16, 3 (2021), 274–306.

[13] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP 2020*. 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[14] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[15] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *NIPS 2020*.

[16] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian J. McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *KDD 2023*. ACM, 1258–1267. https://doi.org/10.1145/3580305.3599519

[17] Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023. Ctrl: Connect tabular and language model for ctr prediction. *arXiv preprint arXiv:2306.02841* (2023).

[18] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *WWW 2022*. ACM, 2320–2329. https://doi.org/10.1145/3485447.3512104

[19] Xiaolong Liu, Liangwei Yang, Zhiwei Liu, Xiaohan Li, Mingdai Yang, Chen Wang, and S Yu Philip. 2023. Group-Aware Interest Disentangled Dual-Training for Personalized Recommendation. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 393–402.

[20] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. In *CIKM 2021*. ACM, 1243–1252. https://doi.org/10.1145/3459637.3482297

[21] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005* (2022).

[22] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[23] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *WWW 2015*. ACM, 111–112. https://doi.org/10.1145/2740908.2742726

[24] Shaoyun Shi, Min Zhang, Xinxing Yu, Yongfeng Zhang, Bin Hao, Yiqun Liu, and Shaoping Ma. 2019. Adaptive Feature Sampling for Recommendation with Missing Content Feature Values. In *CIKM 2019*. ACM, 1451–1460. https://doi.org/10.1145/3357384.3357942

[25] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *ACL 2023*. Association for Computational Linguistics, 1102–1121. https://doi.org/10.18653/v1/2023.findings-acl.71

[26] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems. In *NIPS 2017*. 4957–4966.

[27] Chen Wang, Yueqing Liang, Zhiwei Liu, Tao Zhang, and Philip S Yu. 2021. Pre-training Graph Neural Network for Cross Domain Recommendation. *arXiv preprint arXiv:2111.08268* (2021).

[28] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *KDD 2022*. ACM, 1816–1825. https://doi.org/10.1145/3534678.3539253

[29] Ruoxi Wang, Rakesh Shivanna, Derek Zhiyuan Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed H. Chi. 2021. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. In *WWW 2021*. ACM / IW3C2, 1785–1797. https://doi.org/10.1145/3442381.3450078

[30] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR 2019*. ACM, 165–174. https://doi.org/10.1145/3331184.3331267

[31] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive Learning for Cold-Start Recommendation. In *ACM MULTIMEDIA 2021*. ACM, 5382–5390. https://doi.org/10.1145/3474085.3475665

[32] Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards open-world recommendation with knowledge augmentation from large language models. *arXiv preprint arXiv:2306.10933* (2023).

[33] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]

[34] Lanling Xu, Zhen Tian, Gaowei Zhang, Lei Wang, Junjie Zhang, Bowen Zheng, Yifan Li, Yupeng Hou, Xingyu Pan, Yushuo Chen, Wayne Xin Zhao, Xu Chen, and Ji-Rong Wen. 2022. Recent Advances in RecBole: Extensions with more Practical Considerations.

[35] Liangwei Yang, Zhiwei Liu, Chen Wang, Mingdai Yang, Xiaolong Liu, Jing Ma, and Philip S Yu. 2023. Graph-based alignment and uniformity for recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4395–4399.

[36] Liangwei Yang, Zhiwei Liu, Yu Wang, Chen Wang, Ziwei Fan, and Philip S Yu. 2022. Large-scale personalized video game recommendation via social-aware contextualized graph neural network. In *WWW 2022*. 3376–3386.

[37] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S Yu. 2023. Group identification via transitional hypergraph convolution with cross-view self-supervised learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2969–2979.

[38] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S Yu. 2023. Ranking-based group identification via factorized attention on social tripartite graph. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 769–777.

[39] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S Yu. 2024. Instruction-based Hypergraph Pretraining. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 501–511.

[40] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S Yu. 2024. Unified Pretraining for Recommendation via Task Hypergraphs. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 891–900.

[41] Shenghao Yang, Chenyang Wang, Yankai Liu, Kangping Xu, Weizhi Ma, Yiqun Liu, Min Zhang, Haitao Zeng, Junlan Feng, and Chao Deng. 2023. Collaborative word-based pre-trained item representation for transferable recommendation. In *ICDM 2023*. IEEE, 728–737.

[42] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *KDD 2018*. ACM, 974–983. https://doi.org/10.1145/3219819.3219890

[43] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835* (2023).

[44] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Adapting large language models by integrating collaborative semantics for recommendation. *arXiv preprint arXiv:2311.09049* (2023).

[45] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S. Yu. 2018. Spectral collaborative filtering. In *RecSys 2018*. ACM, 311–319. https://doi.org/10.1145/3240323.3240343

[46] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for New Users and New Items via Randomized Training and Mixture-of-Experts Transformation. In *SIGIR 2020*. ACM, 1121–1130. https://doi.org/10.1145/3397271.3401178