Generative Representational Learning of Foundation Models for Recommendation

Zheli Zhou
zlz114514@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Chenxu Zhu
zhuchenxu1@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Jianghao Lin*
chiangel@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Bo Chen
chenbo116@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Ruiming Tang
tangruiming@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Yong Yu
yyu@apex.sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Weinan Zhang*
wnzhang@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Abstract

Developing a single foundation model with the capability to excel across diverse tasks has been a long-standing objective in the field of artificial intelligence. As the wave of general-purpose foundation models sweeps across various domains, their influence has significantly extended to the field of recommendation systems. While recent efforts have explored recommendation foundation models for various generative tasks, they often overlook crucial embedding tasks and struggle with the complexities of multi-task learning, including knowledge sharing & conflict resolution, and convergence speed inconsistencies. To address these limitations, we introduce **RecFound**, a generative representational learning framework for recommendation foundation models. We construct the first comprehensive dataset for recommendation foundation models covering both generative and embedding tasks across diverse scenarios. Based on this dataset, we propose a novel multi-task training scheme featuring a Task-wise Mixture of Low-rank Experts (TMoLE) to handle knowledge sharing & conflict, a Step-wise Convergence-oriented Sample Scheduler (S2Sched) to address inconsistent convergence, and a Model Merge module to balance the performance across tasks. Experiments demonstrate that RecFound achieves state-of-the-art performance across various recommendation tasks, outperforming existing baselines. The dataset and code are available at https://anonymous.4open.science/r/RecFound-302B.

*Co-corresponding author.

CCS Concepts

• **Information systems → Recommender systems**; **Language models**.

Keywords

Large Language Models, Foundation Models, Recommender Systems

1 Introduction

It has been a long-standing goal to develop a single foundation model to excel across diverse tasks in the field of artificial intelligence [1]. Foundation models, characterized by their extensive knowledge bases and large-scale parameters, have demonstrated remarkable generalization capabilities across various domains, including natural language processing and computer vision. These models rely on large-scale pretraining to capture rich representations, enabling them to adapt to a wide range of downstream tasks with minimal finetuning [39]. Recently, the influence of foundation models has extended to the field of recommendation systems, where they offer unique opportunities to enhance personalization and prediction accuracy by effectively modeling complex user-item interactions and contextual information [17] of distinct sources and formats.

Early efforts in applying foundation models to recommendation systems primarily focus on specific, relatively simple tasks such as tagging [20] and question answering [23]. These approaches often involve adapting general-purpose language models to handle individual recommendation-related tasks, demonstrating the potential of foundation models in this domain. More recent work has aimed to generalize foundation models to handle a broader

spectrum of recommendation tasks [25, 40, 49, 59]. While these works demonstrate promising progress in unifying different recommendation tasks under a single model, they still face the following significant challenges in effectively handling the full complexity of recommendation scenarios.

First, existing approaches overlook the crucial embedding[1] tasks, which are fundamental to recommendation performance. Embedding techniques transform high-dimensional, discrete features such as user and item identifiers or descriptions into low-dimensional, continuous vectors, capturing intricate relationships and facilitating efficient computation. High-quality embeddings are essential for various components of recommendation systems, including retrieval, recall, and visualization [3, 31]. *Second, current models struggle with the complexities of multi-task learning*, particularly in managing the knowledge sharing & conflict and the inconsistent convergence speeds among tasks. These issues can hinder the effective training of foundation models intended to perform well across diverse recommendation tasks, limiting their practical applicability in real-world systems [53].

To address these limitations, we introduce **RecFound**, a generative representational learning framework for recommendation foundation models. Our framework begins with the construction of the first comprehensive dataset that covers both generative and embedding tasks across various aspects of the recommendation domain. Building upon this dataset, we propose a novel multi-task training scheme for recommendation foundation models. From the model perspective, we design a **Task-wise Mixture of low-rank experts (TMoLE)** architecture. TMoLE consists of multiple low-rank experts and a task-oriented router. The router generates a masked routing distribution based on each sample's task to adaptively aggregate the expert outputs. This mechanism allows for effective knowledge sharing between related tasks while mitigating potential conflict among disparate ones by enabling task-specific specialization via the expert integration. From the data perspective, we propose a **Step-wise convergence-oriented sample scheduler (S2Sched)**. S2Sched dynamically adjusts the sampling ratio of different tasks within each training batch based on their non-convergence rates estimated on the validation set. Specifically, it prioritizes sampling from tasks that are converging more slowly, thereby addressing the issue of inconsistent convergence speeds across different recommendation tasks within a single model. Finally, we also introduce a **Model Merge** module to further balance the perfromance across tasks.

Our contributions can be summarized as follows:

- We construct the first comprehensive dataset for recommendation foundation models, covering both 10 generative tasks and 3 embedding tasks across diverse recommendation scenarios.
- We propose RecFound, a novel training framework for recommendation foundation models. We design TMoLE to facilitate effective knowledge sharing & conflict resolution between embedding and generative tasks, S2Sched to address the inconsistency of convergence speeds for different tasks, and the Model Merge module to balance performance across tasks.

[1]In this work, the term "embedding" is interchangeable with "representational".

- Extensive experiments demonstrate that our RecFound framework achieves state-of-the-art performance on a wide range of recommendation tasks, and we provide insightful analysis to support our findings. We release our code, as well as the dataset and model checkpoints, to benefit the research community.

2 Related Works

General Foundation Models in NLP. Extensive research on general foundation models in natural language processing (NLP) has investigated general-purpose methods for addressing a wide range of downstream tasks. To solve various embedding tasks, early influential works focused on learning static word embeddings from large corpora [38, 41]. Subsequent research, such as ELMo [42], BERT [4], and a wave of BERT-family models [32, 47] generates token representations that dynamically reflect their surrounding context, yielding substantial gains in tasks like semantic matching and retrieval. Another line of work investigates foundation models designed for various generative tasks, aiming to solve multiple types of text generation tasks simultaneously. The GPT series [2, 43, 44] demonstrated this by revealing how decoder-only architectures can acquire multitask proficiency only through next-token prediction. More recently, a few studies have explored the possibility of a unified foundation model that can effectively handle both embedding and generative tasks. GritLM [39] trains a large language model to perform both embedding and generative tasks by distinguishing them through instructions. Similarly, UniGen [24] integrates retrieval and question answering into a single model.

Recommendation Foundation Models. Early research on recommendation focused on optimizing single-task generative models [6, 13, 27, 51], such as producing personalized reviews via neural review generators [5] or generating item descriptions from user behavior sequences [58]. These models are typically trained from scratch and lack generalization across tasks. In the wake of large language models, recent works introduce foundation models tailored to e-commerce scenarios [29, 48, 52, 60]. LLaMA-E [49] tunes LLaMA [9] to support various product authoring tasks. EcomGPT [25] leverages generative pretraining over large-scale e-commerce corpora to unify tasks such as product Q&A, summarization, and marketing text generation. eCeLLM [40] further develops a general-purpose e-commerce model by training on high-quality instruction datasets, enabling both in-domain and cross-task generalization. Such efforts demonstrate the growing interest in generative foundation models for recommendation. However, the field still lacks foundation models that support both representation and generative capabilities. To the best of our knowledge, we are the first to develop a unified generative representational learning framework for recommendation foundation models.

3 RecFound Dataset

We present **RecFound**, a meticulously curated dataset that unifies embedding (i.e., representational) and generative tasks within the recommendation domain. Our dataset possesses three key advantages. (1) **Innovation**: We are the first to integrate embedding and generative tasks in recommendation systems, providing a novel benchmark and perspective for research; (2) **Diversity**: The dataset

Table 1: The statistics of our RecFound dataset.

Split	Embedding Tasks			Generative Tasks										Total
	/			General NLP			User Understanding				Item Understanding			
	U2I	Q2I	I2I	QR	AVE	AG	SR	SA	UP	AP	PRP	PM	IP	
Training	86,000	40,000	83,680	3,260	20,000	20,000	20,000	20,000	20,000	15,000	20,000	4,000	15,000	366,940
Validation	3,500	2,600	3,340	500	500	500	500	500	500	500	500	500	500	14,940
Testing	4,800	4,000	4,800	124	1,334	1,332	1,334	1,334	227	1,334	1,333	168	266	22,383

covers multiple recommendation tasks and domains, including e-commerce, social media, music, video streaming, and news recommendations. Each data sample has rich expressiveness to facilitate model training and evaluation; (3) **High-Quality**: All data originates from real user interactions, with rigorous cleaning to remove samples containing illegal symbols, errors, or missing content.

We give a brief introduction to each embedding and generative task below, with detailed data examples provided in Appendix B.

Embedding Tasks include three classical representational learning scenarios for recommendation:

- **User2Item (U2I)**: U2I task involves encoding a user's behavior sequence into an embedding that encapsulates their interests. The output user representation is then utilized to retrieve relevant items. The samples for this task are constructed on Amazon Reviews dataset [12, 14, 36, 55].
- **Query2Item (Q2I)**: In Q2I task, user input queries are embedded to provide precise item recommendations based on the query embeddings. This task is built upon Shopping Queries [46] and ECInstruct datasets [40].
- **Item2Item (I2I)**: I2I task focuses on recommending items similar to a given target item by encoding the embeddings of different item descriptions. All data for this task is derived from Amazon Reviews dataset [12, 14, 36, 55].

Generative Tasks consist of 10 subtasks, which can be categorized into three different types.

- **General NLP**. We use MuDoCo [35], Amazon Reviews [12, 14, 36, 55], and AmazonQA datasets [10], and construct three tasks to provide the model with better language capabilities to solve recommendation questions: (i) Query Rewriting (QR) aims to rewrite obscure queries from users for a better understanding. (ii) Attribute Value Extraction (AVE) expects to extract attribute values of a given item, including titles, descriptions, brands, etc. (iii) Answer Generation (AG) is the common Q&A task with question topics related to recommendation scenarios.
- **User Understanding**: We adopt Amazon Reviews [12, 14, 36, 55] and MovieLens datasets [11], and develop four tasks for a better understanding of users' behaviors, interests, emotions, and intentions. (i) Sequential Recommendation (SR) aims to predict what the user will purchase next given his/her history behavior sequence. (ii) Sentiment Analysis (SA) is a task to estimate the emotion and preference based on users' textual reviews. (iii) User Profile (UP) requires the model to reason and summarize users' detailed preferences based on his/her behavior history. (iv) Answerability Prediction (AP)

judges whether the questions from users can be answered or not.
- **Item Understanding**: We absorb Amazon Reviews [12, 14, 36, 55], Amazon-Google Products [21, 45], and MovieLens [11] datasets to build three tasks aiming at extracting effective information from items. (i) Product Relation Prediction (PRP) conducts three-tier relevance prediction for a given item pair. (ii) Product Matching (PM) judges whether the given two items are the same when they contain different descriptions from different platforms. (iii) Item Profile (IP) requires the model to extract crucial factual information of a given item.

We merge the data samples from all the tasks to form our RecFound dataset, which consists of 3 embedding tasks and 10 generative tasks. Next, we split the dataset into training, validation, and testing sets. We report the detailed dataset statistics in Table 1.

4 Methodology

4.1 General Training Pipeline

Before introducing our specific methods, we first outline the general training pipeline used for both embedding and generative tasks. In our unified framework, all raw input data is transformed into standardized instruction-response pairs. We give example data for each task in Appendix B. Then, these samples are processed via two branches for embedding and generative tasks, respectively.

In the **embedding branch**, batched inputs are passed through a shared LLM backbone and trained using a *contrastive objective* (with InfoNCE loss) to learn a well-structured embedding space. When training backbone is a decoder-only model, we extract the embedding representation for each input sequence by applying mean pooling over the final layer hidden states. In the **generative branch**, samples are similarly batched and optimized using a *token-level cross-entropy loss* to minimize sequence generation errors. Since we aim to establish a unified recommendation foundation model, both branches share the same LLM backbone and parameters, with their respective gradients jointly backpropagated for model update.

However, the shared LLM backbone for multiple heterogeneous embedding and generative tasks inevitably introduces challenges of knowledge sharing & conflict and convergence inconsistency. To address these issues, our RecFound framework incorporates three key modules: Task-wise Mixture of Low-Rank Experts, Step-wise Convergence-Oriented Sample Scheduler, and Model Merge.
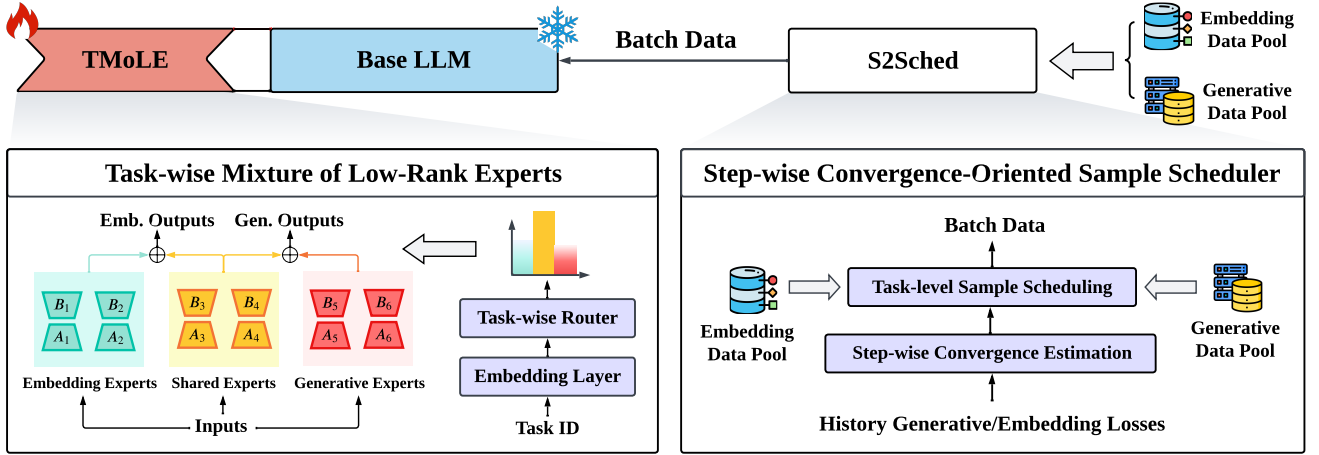
Figure 1: The Method of RecFound. In TMoLE structure, we assume we use 2 embedding experts, 2 shared experts, and 2 generative experts, respectively.

4.2 Task-wise Mixture of Low-Rank Experts

We propose Task-wise Mixture of Low-Rank Experts (**TMoLE**), which integrates multiple low-rank adapters [16] with a task-wise routing mechanism. TMoLE layers are inserted in the multi-head attention projections-specifically, the query, key, value, and output transformations-across each Transformer layer. Each base weight matrix W_0 of one projection is augmented with N parallel low-rank adapters and a **task-wise** router. Different from traditional token-level MoE approaches [7, 34], our router leverages the task type embedding associated with each training sample. Given a task ID i, it is mapped through a learnable embedding table to obtain the corresponding task embedding $\mathbf{e}_i$.

The N low-rank experts are partitioned into three distinct groups: embedding experts $\mathcal{E}$, generative experts $\mathcal{G}$, and shared experts $\mathcal{S}$. Each adapter j comprises trainable matrices $A_j \in \mathbb{R}^{d \times r}$ and $B_j \in \mathbb{R}^{r \times d}$, where $r \ll d$. The task-wise router is implemented as a multi-layer perceptron (MLP) coupled with a masking module. The output gating vector v (i.e., the routing distribution) determines the contribution of each low-rank expert to the final TMoLE output, computed as:

$$v_i = \text{Softmax}(\text{Mask}(\text{MLP}(e_i))). \tag{1}$$

Here, Mask($\cdot$) suppresses irrelevant task dimensions by assigning them a value of $-\infty$, thereby retaining only the experts pertinent to the current task and the shared experts. For example, the corresponding elements of embedding experts in v_i are zero when routing for generative tasks. Notably, the MLP is shared across all the tasks.

The final output of the TMoLE-enhanced projection layer for sample from task i is given by:

$$o = W_0 x + \sum_{j=1}^{N} v_i A_j B_j x. \tag{2}$$

This architecture allows each expert to specialize based on task categories, while the shared experts facilitate cross-embedding-generative task generalization. The router and experts are trained

jointly, promoting end-to-end task-specific adaptation. This design extends previous PEFT-MoE frameworks (e.g., LoRA-MoE [16], TaskMoE [22], and DoRA [30]) and uniquely incorporates task-wise masking and gating over low-rank experts to resolve the problem of knowledge sharing & conflict effectively.

4.3 Step-wise Convergence-Oriented Sample Scheduler

During the training, each task would exhibit different convergence speeds and different levels of difficulty, due to their task-specific characteristics, as well as the inherent knowledge possessed by LLM backbones. Existing works [8, 28] propose adaptive loss reweighting according to the learning dynamics. However, we observe that direct loss reweighting leads to training instability, particularly when tasks use fundamentally different loss functions (i.e., InfoNCE for embedding tasks and Cross Entropy for generative tasks).

Hence, we propose Step-wise Convergence-Oriented Sample Scheduler (**S2Sched**), which estimates the non-convergence rate to schedule the proportion of samples for each task at each training step, rather than simply modifying loss weights. Specifically, we will first perform step-wise non-convergence estimation for each task, and then conduct task-level sample scheduling according to the estimation results, i.e., decreasing the sample ratio of a task if it is converging fast and vice versa.

4.3.1 Step-wise Non-Convergence Estimation. We estimate the non-convergence rate based on the intuition that the validation loss slope of a nearly converged task should approach zero. It is worth noting that we decouple the estimation of embedding and generative tasks to avoid interference caused by differences in loss magnitudes. Hence, we denote the sets of embedding and generative tasks as $\mathcal{T}^{(E)}$ and $\mathcal{T}^{(G)}$, respectively.

For each task type $b \in \{E, G\}$ and task $i \in \mathcal{T}^{(b)}$ at training step t, we can calculate the normalized validation loss slope via linear regression over the last L steps, denoted as $\alpha_i(t)$. Hence, we set a

Table 2: Performance on generative tasks. Average Rank is obtained by calculating the average of the model's rankings on each task. The smaller this value is, the better the comprehensive effect of the model will be. For each task, the best result is in bold, and the second-best result is underlined.

Models	General NLP			User Understanding				Item Understanding			Average Rank↓
	QR↑ (F1)	AVE↑ (F1)	AG↑ (F1)	SR↑ (R@1)	SA↑ (F1)	UP↑ (F1)	AP↑ (F1)	PRP↑ (F1)	PM↑ (F1)	IP↑ (F1)	(Mean (Std))
Baseline Models w/o Finetuning											
Llama3-8B-Instruct	0.926	0.000	0.830	0.000	0.061	0.875	0.420	0.231	0.444	0.880	9.30 (2.57)
Mistral-7B-Instruct	0.936	0.000	0.841	0.005	0.408	0.873	0.738	0.103	0.449	<u>0.892</u>	7.20 (2.71)
Mixtral-8x7B-Instruct	0.948	0.000	0.840	0.005	0.278	0.871	0.647	0.316	0.047	0.885	8.20 (1.94)
Qwen2.5-7B-Instruct	0.952	0.000	0.834	0.027	0.448	0.872	0.383	0.208	0.215	0.881	8.60 (1.80)
Qwen2.5-14B-Instruct	0.944	0.000	0.832	0.063	0.468	0.872	0.518	0.251	0.434	0.886	7.50 (1.43)
Qwen2.5-32B-Instruct	0.961	0.000	0.828	0.128	0.466	0.872	0.488	0.212	0.373	0.887	7.70 (2.24)
GritLM-7B	0.960	0.000	0.840	0.000	0.227	0.868	0.285	0.219	0.491	0.881	8.70 (2.37)
eCeLLM-L	0.943	0.002	<u>0.843</u>	0.177	0.362	0.848	0.812	0.361	**0.994**	0.857	6.70 (3.61)
Baseline Models w/ Finetuning											
eCeLLM	<u>0.973</u>	0.315	0.841	**0.251**	0.532	<u>0.878</u>	0.821	0.450	0.967	0.888	3.20 (1.08)
GritLM	<u>0.973</u>	0.316	0.839	<u>0.242</u>	0.501	0.874	0.814	0.426	0.967	0.889	4.10 (1.51)
Our Models											
RecFound w/o Model Merge	**0.974**	<u>0.341</u>	**0.853**	0.234	**0.545**	**0.895**	<u>0.827</u>	**0.471**	0.988	<u>0.902</u>	1.80 (0.98)
RecFound	<u>0.973</u>	**0.344**	**0.853**	**0.251**	<u>0.533</u>	**0.895**	0.830	<u>0.462</u>	**0.994**	**0.903**	**1.30 (0.46)**

warmup ratio of ς to collect the initial validation loss sequences, i.e., we do not apply non-convergence estimation and sample scheduling at the warmup stage. Based on the step-wise validation loss slopes, we estimate the non-convergence status through the following metrics:

Inter-Task Non-Convergence Rate. We first consider the inter-task convergence status by comparing the tasks of the same type b, i.e., embedding or generative tasks:

$$\gamma_{i,b}^{\text{inter}}(t) = -\text{Softmax}_i\left(|\mathcal{T}^{(b)}| \cdot \frac{\alpha_i(t)}{\sum_{i=1}^{|\mathcal{T}^{(b)}|} |\alpha_i(t)|}\right), \quad (3)$$

where $\alpha_i(t)$ is the normalized validation loss slope of task i estimated over the last L steps. Higher $\gamma_{i,b}^{\text{inter}}(t)$ indicates less convergence of task i at step t in comparison to other tasks.

Intra-Task Non-Convergence Rate. In addition to the inter-task non-convergence among tasks, we also estimate the intra-task non-convergence rate based on the recent L training steps:

$$\gamma_{i,b}^{\text{intra}}(t) = \text{Softmax}_i\left(-L \cdot \frac{\alpha_i(t)}{\sum_{s=t-L+1}^{t} |\alpha_i(s)|}\right). \quad (4)$$

$\gamma_{i,b}^{\text{intra}}(t)$ measures the stability of task i's convergence by comparing the current slope to its own recent history. Higher $\gamma_{i,b}^{\text{intra}}(t)$ indicates less convergence of task i at step t comparison to its history.

Balancing Weight. To further balance the inter- and intra-task non-convergence rate, we calculate the balancing weight as follows:

$$\beta_b(t) = \min\left\{t \cdot \text{Softmax}_t\left(-\frac{\tau t \max_i \alpha_i(t)}{\sum_{k=1}^{t} \max_i |\alpha_i(k)|}\right), \ 1\right\}, \quad (5)$$

where τ is temperature hyperparameter, and $\beta_b(t)$ is shared for all the tasks of type b. Higher $\beta_b(t)$ indicates a balanced relative converging speed among tasks at step t. Lower $\beta_b(t)$ indicates a fairly large disparity in convergence rates across tasks and potential abnormal values estimated by $\gamma_{i,b}^{\text{inter}}(t)$.

Sample Ratio. We interpolate the inter- and intra-task with the balancing weight to calculate the final sample ratio for each task i:

$$\omega_{i,b}(t) = \text{Softmax}_i\left(-\beta_b(t)\,\gamma_{i,b}^{\text{inter}}(t) + \left[1 - \beta_b(t)\right]\gamma_{i,b}^{\text{intra}}(t)\right), \quad (6)$$

where the terms $-\beta_b(t)\,\gamma_{i,b}^{\text{inter}}(t)$ and $\left[1 - \beta_b(t)\right]\gamma_{i,b}^{\text{intra}}(t)$ both convey the positive correlations with the overall non-convergent level for task i of type b at training step t.

4.3.2 Task-level Sample Scheduling. We maintain different batch sizes and schedulers for embedding and generative tasks. For each batch of size $B^{(b)}$ of task type $b \in \{E, G\}$ at step t, the scheduler would re-allocate the sample budget for each task $i \in \mathcal{T}^{(b)}$ according to the noramlized task weight $\omega_{i,b}(t)$. Specifically, the number of samples of task i at step t is $n_i^{(b)}(t) = \lfloor B^{(b)} \cdot w_{i,b}(t) \rfloor$. Then, we would conduct uniform sampling without replacement from the task-wise training data pool to form the final training batch.

In this way, S2Sched adaptively adjusts the task sample ratio per batch within different task types (embedding or generative) at each

Table 3: Performance on embedding tasks. Notations are the same as Table 2.

Models	Item Emb↑ (NDCG@20)	Query Emb↑ (MRR@20)	User Emb↑ (MRR@20)	Average Rank↓ (Mean (Std))
Baseline Models w/o. Finetuning				
gte-Qwen2-7B-Instruct	0.806	0.975	0.431	6.00 (0.00)
SFR-Embedding-Mistral	0.805	0.974	0.419	7.00 (0.00)
bge-large-en-v1.5	0.799	0.957	0.419	8.33 (1.15)
Llama-3-8B-Instruct	0.703	0.438	0.251	10.00 (0.00)
Mistral-7B-Instruct	0.805	0.974	0.419	7.00 (0.00)
GritLM-7B	<u>0.823</u>	0.985	0.458	2.67 (0.58)
eCeLLM-L	0.697	0.197	0.221	11.00 (0.00)
Baseline Models w/ Finetuning				
Embed.Only	0.819	0.983	0.447	4.67 (0.58)
GritLM	0.820	0.985	0.439	4.00 (1.00)
Our Models				
RecFound w/o Model Merge	0.821	<u>0.987</u>	**0.573**	2.00 (1.00)
RecFound	**0.824**	**0.988**	<u>0.570</u>	**1.33 (0.58)**

training step according to real-time validation feedback. Therefore, RecFound can achieve stable and convergence-balanced multi-task learning for recommendation foundation models.

4.4 Model Merge

In multi-task learning scenarios, different model checkpoints often specialize in particular tasks or objectives. This specialization interacts with the inherent seesaw effect of foundation model training, where performance improvements on one task may coincide with regressions on others. To mitigate such interference and consolidate complementary knowledge, we adopt TIES-Merging [56] as the Model Merge module over the learned low-rank experts across different checkpoints. We first remove insignificant parameter updates, then resolve conflicting update directions via majority voting, and finally average the consistent parameters. This approach produces a merged model that better balances the performance across different tasks.

5 Experiments

5.1 Experimental Setup

Datasets. All experiments are conducted on RecFound dataset. The data statistic is reported in Table 1. The prompt templates for generative and embedding tasks are designed with a specific format. We illustrate the prompt template for each task in Appendix B.

Baselines. We first adopt a wide range of non-tuned foundation models as baselines. For generative tasks, we choose Llama-3-8B-Instruct [9], Mistral-7B-Instruct [18], Mixtral-8x7b-Instruct [19], and Qwen-2.5-Instruct series [57](7B,14B,32B). For embedding tasks, we select gte-Qwen2-7B-Instruct [26], SFR-Embedding-Mistral [37], bge-large-en-v1.5 [54], Llama-3-8B-Instruct [9], and Mistral-7B-Instruct [18]. Moreover, we also investigate different finetuning

strategies based on our RecFound dataset. We adopt GritLM [39] as the common finetuned baseline for both generative and embedding tasks, since it is a unified generative representational instruction tuning method. We also choose eCeLLM [40] (tuned on generative tasks) as the finetuned baselines for generative tasks. Since eCeLLM does not support embedding task training, we develop another baseline that discards the generative tasks and is only finetuned on the embedding tasks, i.e., Embed.Only.

Evaluation Metrics. During evaluation, we allow up to 2048 newly generated tokens. In generative tasks, we adopt F1 score as the metric for query rewriting, attribute value extraction, answer generation, user profile, sentiment analysis, answerability prediction, item profile, product relation prediction, and product matching, and use Recall@1 for sequential recommendation. In embedding tasks, we measure the item feature embedding task with NDCG@k, and evaluate user query embedding and user sequence embedding tasks with MRR@k. Following previous works [15, 50], we set $k = 20$.

Implementation Details. Our backbone LLM is Mistral-7B-Instruct [18]. We finetune the backbone LLM on $8 \times$ H100 GPUs for 3 epochs on the RecFound dataset, using a total batch size of 2048 for embedding tasks and 1024 for generative tasks. We use AdamW [33] with a learning rate of 2×10^{-5}. For TMoLE, we apply LoRA [16] to the query/key/value/output projections with rank $r = 16$, $\alpha = 64$, and dropout rate 0.1. We instantiate $N = 6$ low-rank experts for each projection, with the number of embedding experts $E = 2$, generative experts $G = 2$, and shared experts $S = 2$. The task embedding size is 512. For S2Sched, we set the warmup ratio ς as 10%, the length of the history window L as 64 (last steps), and temperature τ as 10. We uniformly sample 128 validation data instances to calculate the normalized validation loss for each task at each training step.

Table 4: Averaged ablation results on different model variants. M.M. stands for the Model Merge module.

Model Variant	Generative (Avg.)	Embedding (Avg.)
RecFound	**0.704**	**0.775**
w/o M.M	0.703	**0.775**
w/o M.M. & S2Sched	0.679	0.762
w/o M.M. & TMoLE	0.678	0.759
w/o M.M. & S2Sched & TMoLE	0.684	0.748

Table 5: Ablation experiments result of generative tasks on different model variants.

Model Variants	General NLP			User Understanding				Item Understanding		
	QR↑ (F1)	AVE↑ (F1)	AG↑ (F1)	SR↑ (R@1)	SA↑ (F1)	UP↑ (F1)	AP↑ (F1)	PRP↑ (F1)	PM↑ (F1)	IP↑ (F1)
RecFound	0.973	**0.344**	**0.853**	**0.251**	0.533	**0.895**	**0.830**	0.462	**0.994**	**0.903**
w/o M.M.	**0.974**	0.341	**0.853**	0.234	**0.545**	**0.895**	0.827	**0.471**	0.988	0.902
w/o M.M. & S2Sched	0.965	0.283	0.852	0.245	0.499	0.885	0.816	0.356	0.988	0.896
w/o M.M. & TMoLE	0.966	0.264	0.847	0.221	0.527	0.887	0.807	0.385	0.976	0.901
w/o M.M. & S2Sched & TMoLE	0.973	0.316	0.839	0.242	0.501	0.874	0.814	0.426	0.967	0.889

Table 6: Ablation experiments result of embedding tasks on different model variants.

Model Variants	Item Emb↑ (NDCG@20)	Query Emb↑ (MRR@20)	User Emb↑ (MRR@20)
RecFound	**0.823**	0.993	**0.510**
w/o M.M.	0.822	**0.995**	0.508
w/o M.M. & S2Sched	0.820	0.992	0.474
w/o M.M. & TMoLE	0.822	0.991	0.465
w/o M.M. & S2Sched & TMoLE	0.820	0.985	0.439

5.2 Main Results

We evaluate our proposed RecFound and other baseline models. The results of generative and embedding tasks are reported in Table 2 and Table 3, respectively. We can observe that RecFound achieves the best performance across most generative tasks and embedding tasks, outperforming both larger-scale LLMs and strong instruction-tuned baselines. This demonstrates the effectiveness of TMoLE and S2Shed, which helps resolve the knowledge sharing & conflict and convergence inconsistency problems for multi-task foundation model training. In addition, the model merge module further improves the performance of RecFound for both generative and embedding tasks on average. This indicates that the model merge module combines the advantages of multiple model checkpoints, compensating for their limitations and showing excellent compatibility with heterogeneous multi-task learning.

5.3 Ablation Study

We conduct ablation studies by removing the Model Merge, S2Shed, and TMoLE components. The average results on generative and embedding tasks are reported in Table 4, while detailed results on each task are given in Table 5 and Table 6. From Table 4, we can obtain the following observations:

- Removing the model merge module only results in slight performance degradation. While the model merge module leads to performance gains on some tasks and declines on others, it contributes to an overall improvement in model performance as shown in Table 2 and Table 3.
- When removing S2Shed, RecFound reverts to uniform sample scheduling among tasks. This leads to an imbalance of overfitting on simple tasks and underfitting on more challenging ones, thereby leading to inferior performance.

Averaged similarity for intra- and inter-class tasks.

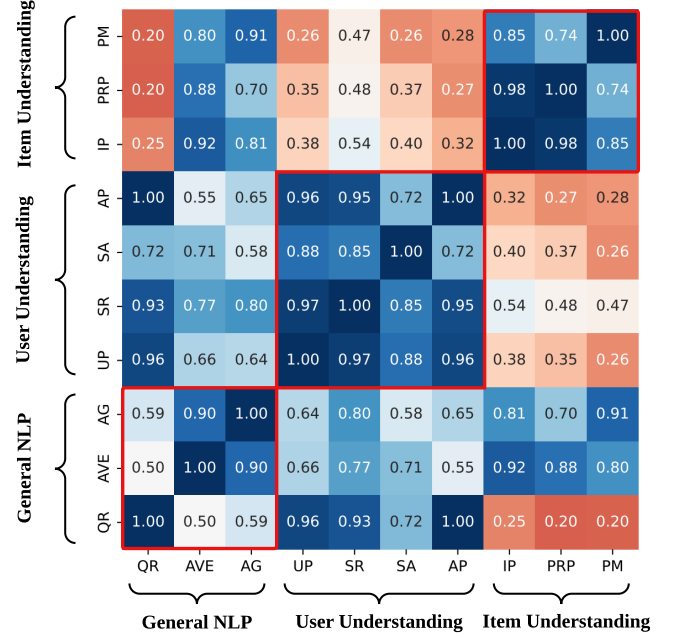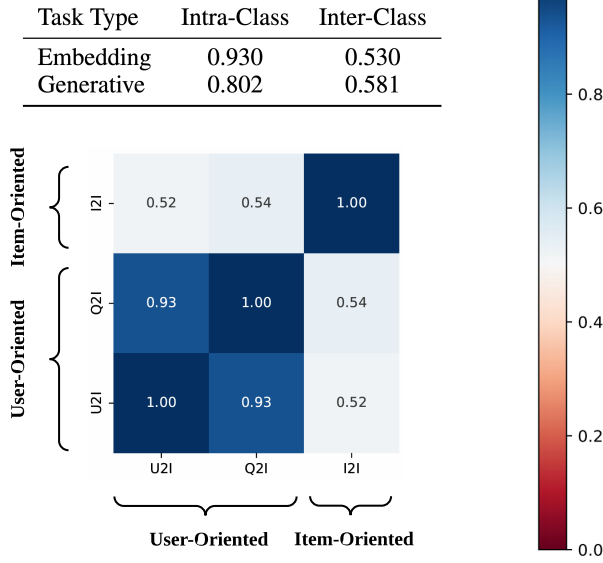| Task Type	Intra-Class	Inter-Class
Embedding	0.930	0.530
Generative	0.802	0.581

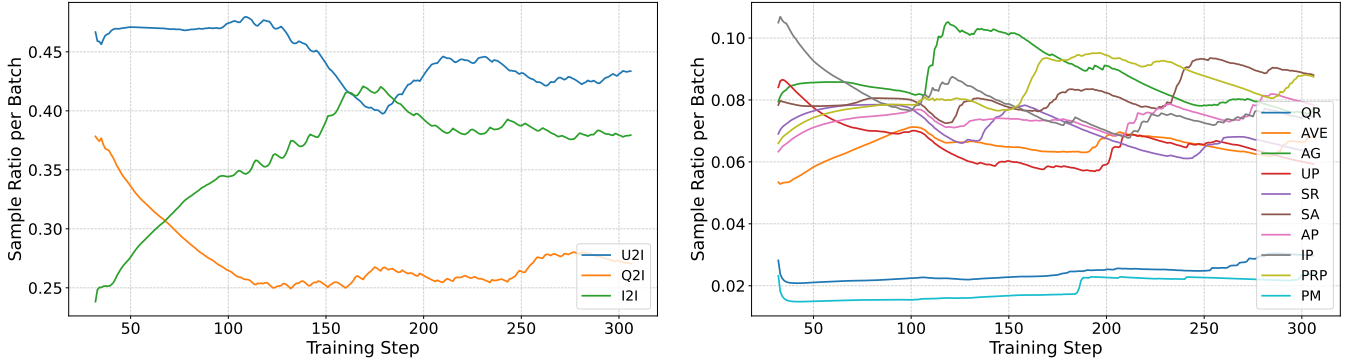Figure 2: Similarity heatmaps of the routing distribution across different tasks.

Figure 3: The evolving sample ratio per batch of different tasks during the training. The training step starts from 32 due to the warmup progress of S2Sched.

- Removing TMoLE reduces the learning structure of Rec-Found into a common LoRA [16] module. This impairs the knowledge sharing and conflict resolution, which in turn degrades both generative and embedding performance significantly.

These results confirm that Model Merge, S2Shed, and TMoLE are all essential for knowledge-disentangled and convergence-balanced multi-task learning of recommendation foundation models.

Table 5 and Table 6 also present the detailed results of the ablation study on generative and embedding tasks, respectively. The trends observed in the detailed task-level metrics are largely consistent with the averaged results, with performance gradually declining as more components are removed.

5.4 In-Depth Analysis

5.4.1 Routing Distributions Among Tasks. To investigate whether the router in TMoLE learns domain-specific characteristics for task-wise knowledge sharing and conflict, we visualize the similarity heatmap of the routing distribution v_i in Eq. (1) across different tasks. Specifically, we compute the pairwise cosine similarity between each embedding or generative task pair, and illustrate the heatmap in Figure 2.

We observe that tasks with higher similarity (i.e., intra-class denoted by the red box) tend to share more similar routing distributions, whereas tasks that differ substantially-particularly those from different classes-exhibit more divergent routing behaviors. This suggests that TMoLE can successfully capture both the commonalities

and distinctions across tasks, thereby facilitating effective embedding & generative task learning for recommendation foundation models.

5.4.2 Scheduled Sample Ratio per Batch. To investigate how S2Sched works to balance the convergence speeds of different tasks, we visualize the sample ratio of different tasks per batch calculated in Eq. (6) w.r.t. the training steps in Figure 3. We observe that, during the training, the sample ratios of all tasks exhibit a general pattern of rising and falling alternately over time, rather than monotonically increasing/decreasing or remaining stable. A task that is approaching convergence will be assigned a lower weight, but later its weight will be increased again to prevent the forgetting problem of model learning. Such behavior reflects S2Sched's ability to adaptively modulate sampling priorities across tasks, promoting more balanced optimization when tasks converge at different rates.

6 Conclusion and Limitation

In this work, we present RecFound, a unified generative representational learning framework for recommendation foundation models. We construct the first comprehensive dataset for recommendation foundation models covering 3 embedding tasks and 10 generative tasks. RecFound addresses the problems of knowledge sharing & conflict and convergence imbalance through the proposed TMoLE, S2Sched, and Model Merge modules. Extensive experiments demonstrate strong generalization and task adaptability of RecFound model, highlighting the potential of unified recommendation foundation models for both generative and embedding tasks. While effective, RecFound's performance may be sensitive to task-specific data imbalance and does not yet support continual task expansion.

References

[1] Rishi Bommasani, Drew A Hudson, Eleanor Adlam, Atticus von Werra, Zachary Mahoney, Aran Neeraj, Eric Roberts, Stephen Casais, Roger Mouawad, Joel Batten, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, and et al. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165* (2020). https://arxiv.org/abs/2005.14165

[3] Li Chen et al. 2022. Deep Variational Embedding Representation on Neural Collaborative Filtering for Recommender Systems. *Applied Sciences* 12, 9 (2022), 4168.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186. https://arxiv.org/abs/1810.04805

[5] L. Dong, Y. Shi, and T. Liu. 2017. Neural Review Generation for Personalized Recommendations. In *CIKM*.

[6] Lifu Dong, Tianbao Zhang, Siwei Xu, Yun Li, Shengqiang Liu, Jun Guo, Yi Zhu, and Kewei Wang. 2017. Sequence-to-sequence learning for review generation and recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1109–1112.

[7] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LoRAMoE: Alleviate World Knowledge Forgetting in Large Language Models via MoE-Style Plugin. arXiv:2312.09979 [cs.CL] https://arxiv.org/abs/2312.09979

[8] Zi Gong, Hang Yu, Cong Liao, Bingchang Liu, Chaoyu Chen, and Jianguo Li. 2024. CoBa: Convergence Balancer for Multitask Finetuning of Large Language Models. arXiv:2410.06741 [cs.CL] https://arxiv.org/abs/2410.06741

[9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv e-prints* (2024), arXiv–2407.

[10] Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C Lipton. 2019. Amazonqa: A review-based question answering task. *arXiv preprint arXiv:1908.04364* (2019).

[11] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. doi:10.1145/2827872

[12] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[13] Balázs Hidasi, Alexandros Karatzoglou, Dalius Baltrunas, and Dániel Tikk. 2015. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*.

[14] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[15] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[17] Chengkai Huang, Tong Yu, Kaige Xie, Shuai Zhang, Lina Yao, and Julian McAuley. 2024. Foundation Models for Recommender Systems: A Survey and New Perspectives. *arXiv preprint arXiv:2402.11143* (2024).

[18] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[19] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. arXiv:2401.04088 [cs.LG] https://arxiv.org/abs/2401.04088

[20] Anand Konjengbam, Nagendra Kumar, and Manish Singh. 2020. Unsupervised tag recommendation for popular and cold products. *Journal of Intelligent Information Systems* 54 (2020), 545–566. doi:10.1007/s10844-019-00574-9

[21] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 484–493.

[22] Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021. Beyond Distillation: Task-level Mixture-of-Experts for Efficient Inference. arXiv:2110.03742 [cs.CL] https://arxiv.org/abs/2110.03742

[23] Tuan Manh Lai, Trung Bui, Nedim Lipka, and Sheng Li. 2019. Supervised transfer learning for product information question answering. *arXiv preprint arXiv:1901.02539* (2019).

[24] Xiaoxi Li, Yujia Zhou, and Zhicheng Dou. 2023. UniGen: A Unified Generative Framework for Retrieval and Question Answering with Large Language Models. (2023). arXiv:2312.11036.

[25] Yangning Li, Shirong Ma, Xiaobin Wang, Shen Huang, Chengyue Jiang, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. [n. d.]. EcomGPT: Instruction-Tuning Large Language Models with Chain-of-Task Tasks for E-commerce. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press. doi:10.1609/aaai.v38i17.29820

[26] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281* (2023).

[27] Dawen Liang, Rahul Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*. 689–697.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[29] Haoxi Liu, Yunjun Zhang, Xiaochen Yu, Xinxing Wang, Peng Cao, Jian Zhang, Yu Chen, and Haitao Huang. 2023. FIM: A Fine-grained and Multi-aspect Recommendation Model based on Large Language Model. *arXiv preprint arXiv:2308.04571* (2023).

[30] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. DoRA: Weight-Decomposed Low-Rank Adaptation. arXiv:2402.09353 [cs.CL] https://arxiv.org/abs/2402.09353

[31] Yuxin Liu et al. 2023. Embedding in Recommender Systems: A Survey. *arXiv preprint arXiv:2310.18608* (2023).

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Li, Artem Kochanov, Omer Mahoney, Piotr Dollar, Veselin Stoyanov, et al. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[33] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[34] Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. 2024. MoELoRA: Contrastive Learning Guided Mixture of Experts on Parameter-Efficient Fine-Tuning for Large Language Models. doi:10.48550/arXiv.2402.12851 arXiv:2402.12851 [cs.CL]

[35] Scott Martin, Shivani Poddar, and Kartikeya Upasani. 2020. MuDoCo: Corpus for Multidomain Coreference Resolution and Referring Expression Generation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association, Marseille, France, 104–111. https://aclanthology.org/2020.lrec-1.13/

[36] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[37] Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. SFR-Embedding-Mistral: Enhance Text Retrieval with Transfer Learning. Salesforce AI Research Blog. https://www.salesforce.com/blog/sfr-embedding/

[38] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *ICLR Workshop*. https://arxiv.org/abs/1301.3781

[39] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative Representational Instruction Tuning. arXiv:2402.09906 [cs.CL] https://arxiv.org/abs/2402.09906

[40] Bo Peng, Xinyi Ling, Ziru Chen, Huan Sun, and Xia Ning. [n. d.]. eCeLLM: Generalizing Large Language Models for E-commerce from Large-scale, High-quality Instruction Data. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR. https://proceedings.mlr.press/v235/peng24c.html

[41] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*. 1532–1543. doi:10.3115/v1/D14-1162

[42] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. *arXiv preprint arXiv:1802.05365* (2018). https://arxiv.org/abs/1802.05365

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. *OpenAI Blog* (2018). https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[44] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Technical Report* (2019). https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[45] Rahm, Erhard. 2010. Benchmark datasets for entity resolution. https://dbs.uni-leipzig.de/research/projects/benchmark-datasets-for-entity-resolution.

[46] Chandan K Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping queries dataset: A large-scale ESCI benchmark for improving product search. *arXiv preprint arXiv:2206.06588* (2022).

[47] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 3982–3992.

[48] Shijie Shen, Yunjia Ma, Jian Retrieval Zhang, Zhuoyi Zhang, Zihan Wang, Liangkun Li, Shuyu Deng, Yunfan Zhang, Jiaxing Zhang, Shaohan Jiang, et al. 2023. Large Language Models are Zero-Shot Recommenders. *arXiv preprint arXiv:2301.00794* (2023).

[49] Kaize Shi, Xueyao Sun, Dingxian Wang, Yinlin Fu, Guandong Xu, and Qing Li. 2024. LLaMA-E: Empowering E-commerce Authoring with Object-Interleaved Instruction Following. arXiv:2308.04913 [cs.CL] https://arxiv.org/abs/2308.04913

[50] Yan-Martin Tamm, Rinchin Damdinov, and Alexey Vasilev. 2021. Quality metrics in recommender systems: Do we calculate metrics consistently?. In *Proceedings of the 15th ACM conference on recommender systems*. 708–713.

[51] Jun Wang, Lantao Yu, Xiang Zhang, Yunhai Gong, Yingguo Xu, Bo Wang, Shi Zhang, Zhongfei Zhang, Yongfneg Li, and Chen Haifeng. 2017. IRGAN: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 204–213.

[52] Xinxing Wang, Haoxi Liu, Nan Xu, Xiaochen Yu, Ruiming Wu, Yunjun Zhang, Jian Zhang, Haitao Huang, and Yu Chen. 2024. UniRec: Unified and Fine-grained Recommendation with Large Language Model. *arXiv preprint arXiv:2401.14614* (2024).

[53] Yan Wu et al. 2023. Robust Training Objectives Improve Embedding-Based Retrieval in Industrial Recommendation Systems. *arXiv preprint arXiv:2409.14682* (2023).

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

[55] Huimin Xu, Wenting Yang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up Open Tagging from Tens to Thousands: Comprehension Empowered Attribute Value Extraction from Product Title. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5214–5223. doi:10.18653/v1/P19-1514

[56] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems* 36 (2023), 7093–7115.

[57] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115* (2024).

[58] X. Yang, Y. Zhang, and B. Li. 2018. Behavior-Driven Item Description Generation. In *WWW*.

[59] Shuo Zhang, Boci Peng, Xinping Zhao, Boren Hu, Yun Zhu, Yanjia Zeng, and Xuming Hu. 2024. LLaSA: Large Language and E-Commerce Shopping Assistant. https://arxiv.org/abs/2408.02006

[60] Gaojie Zhou, Yupeng Xu, Wayne Xin Ding, Ying Wang, Zijian Zhang, Longxing Wang, Tianyu Yao, Yujie Zhu, Jiaqi Song, Zhengyu Bi, et al. 2024. Large Language Models for Recommendation: A Survey. *arXiv preprint arXiv:2401.12397* (2024).

A Broader Impact

As a unified generative representational learning framework, Rec-Found promises to enhance personalization by capturing richer user–item interactions and broader tasks from various recommendation domains. It can improve recommendation quality, user satisfaction, and economic inclusivity for small businesses and content providers. However, recommendation foundation models can inadvertently amplify biases-leading to unfair exposure of items or user groups-and raise serious privacy concerns, as sensitive attributes may be inferred from learned embeddings. To mitigate these risks, we recommend adopting privacy-preserving techniques such as differential privacy or federated learning during data collection and model finetuning. Embedding human-in-the-loop monitoring and user feedback mechanisms can further detect and correct unintended biases or misleading generative outputs, steering RecFound toward ethical and equitable recommendation practices.

B Data Construction Details

B.1 Prompt Templates

The following prompts are used for evaluating general-purpose recommendation LLMs on generative tasks in RecFound Datasets. For Mistral-7B Instruct-v0.3, the prompt is wrapped with "[INST]". For Llama-3-8B-Instruct and Qwen-2.5-Instruct series, the prompt is wrapped with "<im_start>" and "<im_end>". We conduct zero-shot evaluations by default, so there's no one-shot or few-shot prompt template.

Prompt Template with Options for Generative Tasks.
- System prompt: Below is an instruction that describes a task. Write a response that appropriately completes the request.
- Instruction: {instruction}
  - input: {input}
  - options: {options}
  - response:

Prompt Template without Options for Generative Tasks.
- System prompt: Below is an instruction that describes a task. Write a response that appropriately completes the request.
- Instruction: {instruction}
  - input: {input}
  - response:

To train and evaluate embedding tasks, we need to add special tokens "<|embed|>" as bos token for embedding inputs. An embed EOS is useless as there is no generative loss on it, so it won't be learned. There is also no need to design a prompt template for embedding tasks, we only need to extract embedding vectors from the mean pooling of last-hidden states.

B.2 Diverse Instructions

For each embedding and generative task, we design several instructions to help LLMs understand various user requests. For each task, we show all the instructions we designed below.

Query Rewriting.
- Please undertake the task of rephrasing the questions and responses in a given conversation. You will be given a dialogue

between two individuals (I will use A: for the first person and B: for the second person). For each query/response, an indication will be provided in parentheses at the end showing whether it requires rewriting. If no rewriting is needed, simply return the original content; if it does, use the preceding context to modify the sentence appropriately, making its meaning clearer, such as by specifying pronouns, adding place names or times, etc.
- Please carry out a task to rephrase the queries and replies in a given conversation. You will receive a dialogue between two people (I will label the first person as A: and the second person as B:). For each query/reply, I will note in parentheses at the end whether the sentence needs rewriting. If no rewriting is required, simply return the original text; if rewriting is required, use the preceding context to modify the sentence appropriately to make its meaning clearer, such as specifying pronouns, adding locations or times, etc.
- Please accomplish the task of rephrasing the queries and responses in a given conversation. You will be provided with a dialogue between two individuals (I will use A: to refer to the first person and B: to refer to the second person). For each query/response, I will specify in parentheses at the end whether it needs to be rewritten. If no rewriting is needed, return the original content; if rewriting is required, use the previous context to appropriately modify the sentence to make its meaning clearer, such as specifying pronouns, adding place names, or times, etc.
- Please perform the task of rephrasing the questions and answers in a given conversation. You will receive a dialogue between two people (denoted as A: for the first person and B: for the second person). For each question/answer, I will indicate at the end in parentheses whether it needs to be rewritten. If no rewriting is necessary, simply return the original content; if rewriting is necessary, use the context to modify the sentence appropriately to make its meaning clearer, such as by specifying pronouns, adding place names, or times, etc.
- Please complete a task of rewriting the queries and replies in a given conversation. You will receive a dialogue between two people (I will use A: to denote the first person and B: to denote the second person). For each query/reply, I will indicate in parentheses at the end whether the sentence needs to be rewritten. If rewriting is not required, you just need to return the original content; if rewriting is required, please use the preceding context to modify the sentence appropriately to make its meaning clearer, such as specifying pronouns, adding place names or times, etc.

Attribute Value Extraction.
- Given the title, description, feature, price, and brand of a product and a set of target attributes, extract the value of each target attribute from the product information. Output the extracted value and the corresponding source (e.g., title or feature) denoting where the value is extracted.
- Extract the value of the target attribute from the given product information and output it along with the corresponding source.

- Parse the product information to locate the target attribute, and then provide the extracted value of the target attribute and its source in the output, specifying None if the attribute is not present.
- First, identify the target attributes from the provided list. Then, scan the product title, description, feature, and brand to extract the values associated with each target attribute. Finally, create a list of dictionaries, each containing the extracted attribute, its corresponding value, and the source where it was found.
- Using the product's title, description, features, price, and brand, identify and retrieve the values associated with a specified set of target attributes. Output the extracted values along with their respective sources (e.g., title or feature) indicating where each value was found.

Answer Generation.

- Given a question and the related document, and generate the answer to the question based on the information provided in the document.
- Generate an answer to the question by utilizing the information contained in the document.
- Extract information from the supporting document to answer the given question.
- Answer the given question using the supporting document.
- Answer the given question by extracting information from the supporting document.

User Profile.

- Given a [Male/Female] user who is aged 18-24 and in a [student/service staff/.../unknown occupation], this user's movie viewing history over time is listed below. [History Sequence] Analyze user's preferences on movies (consider factors like genre, director/actors, time period/country, character, plot/theme, mood/tone, critical acclaim/award, production quality, and soundtrack). Provide clear explanations based on relevant details from the user's movie viewing history and other pertinent factors.
- Given user's book rating history: [List of Bookname: Rating] Analyze user's preferences on books about factors like genre, author, writing style, theme, setting, length and complexity, time period, literary quality, critical acclaim (Provide clear explanations based on relevant details from the user's book viewing history and other pertinent factors.)

Sequential Recommendation.

- Given the products the user has purchased in history, rank the items in the listed options and output the item that the user is most likely to purchase next. Answer from one of the options.
- Based on the user's historical purchases, rank the items in options and predict the next product of the user's interest from the provided options.
- Rank the items in options and predict the user's next purchase from the listed options by analyzing her historical purchases.
- The user's purchase history implies her preferences. Rank the items in the options based on the user's preferences.

Output the item that the user is most likely to purchase next from the options.
- Rank items in listed options based on the user's purchase history to determine the item that the user is most likely to purchase next. Output the item with the highest likelihood of being the next purchase.

Sentiment Analysis.

- Given the user's review, identify the user's sentiment from the listed options. Answer using one of the options.
- Assess the user's sentiment in the provided review and select the appropriate sentiment option from the list as the answer.
- Determine the sentiment expressed by the user in her review from the provided choices, and respond by selecting one of the available options.
- Carefully assess the user's review for any strong expressions of sentiment, either positive or negative. Based on your analysis, select the most fitting sentiment option from the provided list as output.
- Analyze the user's review text and determine the overall sentiment expressed, then choose the corresponding sentiment option from the provided list (A: very positive, B: positive, C: neutral, D: negative, E: very negative) based on the identified sentiment.

Answerability Prediction.

- Given a question and the related document, predict if the question is answerable based on the information provided in the document. Output only yes or no.
- Evaluate the answerability of a question by analyzing the related document, outputting yes if the document contains information addressing the question, and no otherwise.
- Analyze a question and its supporting document. Predicting answerability based on the information provided in the document. Output yes if the document contains relevant information to answer the question, otherwise output no.
- Given a question and its related document, determine if the question is answerable by analyzing the information in the document. Output yes if the document addresses the question, or no otherwise.
- Output yes if the supporting document can answer the given question. Otherwise, output no.

Item Profile.

- Introduce movie [Movie name] and describe its attributes (including but not limited to genre, director/cast, country, character, plot/theme, mood/tone, critical acclaim/award, production quality, and soundtrack).
- Introduce book [Book name], which is from [Source] Page and describe its attributes including but not limited to genre, author, writing style, theme, setting, length and complexity, time period, literary quality, critical acclaim.

Product Relation Prediction.

- Given the title of two products, predict if the two products are similar, if the two products will be purchased or viewed together. Answer only from the options.

- Analyze the titles of Product 1 and Product 2 to determine if they are similar, if they will be purchased or viewed together, and choose the corresponding option.
- Evaluate the titles of Product 1 and Product 2, then choose the option that best describes the relation between the two products.
- Evaluate the titles of Product 1 and Product 2 to assess their similarity and whether they are likely to be purchased or viewed together. Then, select the appropriate option.
- Predict whether two products are similar, whether two products are likely to be purchased or viewed together based on their titles. Choose your answer from the provided options.

Product Matching.

- Given the title, description, manufacturer, and price of two products, identify if they are the same product. Only output yes or no.
- Analyze the title, description, manufacturer, and price between the two products below and generate an output of yes if the two products are the same, otherwise respond with no.
- Check the details of the two products to see if they refer to the same product. Output only yes or no.
- Based on the product information, predict if the two products are identical or not. Output yes if they are identical or no otherwise.
- Compare the details of two given products to determine if they are identical. Output yes if they are identical or no otherwise.

User2Item.

- Represent this sequence of user actions to predict the next product he/she will purchase (User sequence is in reverse chronological order):
- Encode this user activity sequence to forecast their next product purchase (User sequence is in reverse chronological order):
- Create an embedding of this user sequence to determine the next product they will purchase (User sequence is in reverse chronological order):
- Generate a representation of this user sequence to find the next product this user will buy (User sequence is in reverse chronological order):
- Represent this user sequence to retrieve the next product this user will buy (User sequence is in reverse chronological order):

Query2Item.

- Represent this user query to find a matching item:
- Generate a representation of this user query to locate a relevant item:
- Create an embedding of this user query to retrieve a relevant item:
- Use the representation of the user query to identify a relevant item:
- Represent this user query to retrieve a relevant item:

Item2Item.

- Create an embedding for the product using its title and categories to help find similar items
- Generate an embedding for a product based on its title and categories to facilitate the retrieval of similar items item Based on the title and categories of a product, generate an embedding to identify similar products
- Using the product's title and categories, produce an embedding to retrieve items that are alike
- Given the title of a product and its category, please generate an embedding for the product to retrieve similar items

B.3 Data Examples

In this section, we provide an example for each task, with complete instructions (including system prompt and user prompt) and the corresponding ground-truth.

Query Rewriting.

- **Instruction:**
  Please perform the task of rephrasing the questions and answers in a given conversation. You will receive a dialogue between two people (denoted as A: for the first person and B: for the second person). For each question/answer, I will indicate at the end in parentheses whether it needs to be rewritten. If no rewriting is necessary, simply return the original content; if rewriting is necessary, use the context to modify the sentence appropriately to make its meaning clearer, such as by specifying pronouns, adding place names, or times, etc. Now the dialogue you need to rewrite is shown below:
  A: Do I have missed calls today? (rewrite required: False)
  B: Yes, you have two. It's from Don and Ezra. (rewrite required: True)
  A: Call Ezra first. (rewrite required: False)
  B: Calling her now. (rewrite required: True)
  Based on the given dialogue, your rewritten result is:
- **Answer:**
  A: Do I have missed calls today?
  B: Yes, you have two missed calls today. It's from Don and Ezra.
  A: Call Ezra first.
  B: Calling Ezra now.

Attribute Value Extraction.

- **Instruction:**
  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
  –Instruction:
  Given the title, description, feature, price, and brand of a product and a set of target attributes, extract the value of each target attribute from the product information. Output the extracted value and the corresponding source (e.g., title or feature) denoting where the value is extracted.
  –Input:
  Product title: Bencore Multi Functional Molle Tactical Messenger Bag.

Product description: This rugged/durable tactical shoulder bag provides perfect and stylish solution for almost any scenario. The bag is made of durable nylon construction that will not tear, color will not fade. The bag has many MOLLE straps through the bag for all your MOLLE accessories. The bag contains many roomy compartments as pictured and comes in many stylish colors. Design, comfort and functionality was the emphasis of this bag which is why we made sure the bag is fully ergonomic, lightweight and has many roomy pockets and Velcro patches throughout the bag. The product comes with the Bencore Life Time warranty and is satisfaction guaranteed. Bencore is a leading manufacturer in outdoor Apparel/Accessories, from Par cords to Backpacks to basic outdoor essentials.
Product feature: Durable heavy-duty, lightweight Nylon construction, will not tear or break even under extreme conditions - Lifetime Warranty, Rugged, roomy main drawstring-closed compartment provides secure storage space for your gear; MOLLE System, works with most MOLLE accessory, Front pocket provides quick access, roomy interior pocket for convenient separated storage, concealed back pocket with zipper closure, Padded and fully ergonomic System, adjustable shoulder strap for comfortable handling.
Product brand: Bencore
Target attributes: Material
–Response:

- **Answer:**
Attribute: material; Value: nylon; Source: product description
Attribute: material; Value: nylon; Source: product feature.

Answer Generation.

- **Instruction:**
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
–Instruction:
Generate an answer to the question by utilizing the information contained in the document.
–Input:
Would this be considered a dark metal pan which requires a lower baking temp than shiny pans?"
Document:
I've been baking all our bread for several years and needed new pans. These are very nice, a good weight, smooth finish, easy to clean, and the bread bakes perfectly. Would definitely recommend.
Great for baking bread. Bread slides out of Non-stick pan easily. Definitely would recommend this for baking. Sturdy design. I will be purchasing another one soon.", "I kept buying bread pans as my recipes called for an 8 by 4 bread pan. They would arrive and I would measure them and they were never four inches deep. Then I finally realized they were talking about width.
These are good metal pans just as the ad showed. There are not any flaws or other negatives. thanks, Mike", "I needed to purchase an 8x4 loaf pan for baking gluten free bread. It has been wonderful! My family is able to enjoy sandwiches

again. The bread releases well and the pans wash easily. I do spray them with cooking spray before baking.
–Response:

- **Answer:**
The pan is a dark metal, but I use the same temp as for lighter pans and bake until the top of my bread is golden brown and the crust within the pan seems to be just fine.

User Profile.

- **Instruction:**
Below is an instruction that describes a task. Write a response that appropriately completes the request.
–Instructions:
Given a female user who is aged 50-55 and self-employed, this user's movie viewing history over time is listed below. Sabrina (1995), 4 stars; Sabrina (1954), 5 stars; Gone with the Wind (1939), 5 stars; Citizen Kane (1941), 4 stars; You've Got Mail (1998), 3 stars; Thelma & Louise (1991), 3 stars; American Beauty (1999), 4 stars; Broadway Melody, The (1929), 4 stars; American President, The (1995), 4 stars; Corrina, Corrina (1994), 4 stars; Analyze user's preferences on movies (consider factors like genre, director/actors, time period/country, character, plot/theme, mood/tone, critical acclaim/award, production quality, and soundtrack). Provide clear explanations based on relevant details from the user's movie viewing history and other pertinent factors.
–Response:

- **Answer:**
Based on the user's movie viewing history, it seems that she enjoys classic films from the 1930s and 1940s, as well as more recent dramas and romantic comedies. She seems to appreciate films with strong female characters, as evidenced by her ratings of Thelma & Louise and Sabrina (both versions). She also appears to value critical acclaim and award-winning films, as she gave high ratings to Gone with the Wind and Citizen Kane. Additionally, she enjoys films with good production quality, as indicated by her positive rating for The American President.
It is possible that the user has a preference for films with a romantic or sentimental tone, as her ratings for You've Got Mail, Corrina, Corrina, and the two versions of Sabrina are all relatively positive.
Based on this information, it could be recommended that the user explore classic films from the 1930s and 1940s, as well as critically acclaimed dramas and romantic comedies. She may also enjoy films with strong female characters and a sentimental tone.

Sequential Recommendation.

- **Instruction:**
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
–Instruction:
Given the products the user has purchased in history, rank the items in the listed options and output the item that the

user is most likely to purchase next. Answer from one of the options.

1st: Wilson Pro feel Tennis Vibration Dampener. Sports & Outdoors. Sports & Fitness. Wilson.

2nd: COSMOS 5 PCS White cotton sports basketball head-band / sweatband head sweat band/brace. Sports & Outdoors. Sports & Fitness. Cosmos.

3rd: Gamma Guard. Sports & Outdoors. Sports & Fitness. Gamma.

4th: SummerHouse 1 Gram Per Inch Golf High Density Lead Tape Weight Self-Adhesion for Driver Fairway Hybrid Iron Clubs Head - 1/2"x 100". Sports & Outdoors...

–Options:

A: Aceson A7 7Quad Core Google Android 4.4 Tablet PC, IPS Display, 8GB Storage, Bluetooth 4.0, Dual Camera, Support Netflix, Skype, Facebook, YouTube, Google...

B: Pinzon Heather Flannel Blanket, Full/Queen, Blue Heather. Home & Kitchen. Bedding. Pinzon by Amazon

C: Sony LCS-SC5 Alpha System Carrying Case. Electronics. Camera & Photo. Sony.

D: Advent Candles Set 4, 10" X 3/4". Home & Kitchen. Home Dcor. Abbey Gift.

E: Nidec Servo Gentle Typhoon D1225C12B5AP 120mm 1850 rpm Silent Case Fan. Electronics. Computers & Accessories. SERVO.

F: FYY Case for iPad Air 2 - Premium PU Leather Case Smart Auto Wake/Sleep Cover with Hand Strap, Card Slots, Pocket for iPad Air 2.

–Response:

- **Answer:**
  C

Sentiment Analysis.

- **Instruction:**
  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
  –Instruction:
  Given the user's review, identify the user's sentiment from the listed options. Answer using one of the options.
  –Input:
  This gives me control of the volume while on a treadmill and I don't have to stop and go back to the stereo or TV while I'm walking.
  Options:
  A: very positive
  B: positive
  C: neutral
  D: negative
  E: very negative
  –Response:
- **Answer:**
  A: very positive

Answerability Prediction.

- **Instruction:**

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

–Instruction:

Given a question and its related document, determine if the question is answerable by analyzing the information in the document. Output yes if the document addresses the question, or no otherwise.

–Input:

question: Is the lamp base weighted well? Will the lamp tip or lean over if you adjust the light to any angle or extend the arm out to it's maximum capacity?,

document: This is a wonderful light. It provides exactly the kind of light I need for the close hand work I do. I highly recommend this light!

This is my second lamp of this style and model. i purchased the first one locally for my husband several months ago. I paid almost $60.00 more for that one. It was so helpful for him, reducing eye strain and thus allowing him to read more, that I decided to try to find another one for myself. I thought I was OK reading with just the regular lamps in our living room until I sat under my husband's Ott Lite. What a difference it made!

I understand it is an investment, but you only have one pair of eyes and the older you get, the harder it is to focus without eye strain and the consequential headaches that can accompany poor lighting. I'm glad I purchased these lamps and highly recommend them to anyone who loves to read.

–Response:

- **Answer:**
  No

Item Profile.

- **Instruction:**
  Below is an instruction that describes a task. Write a response that appropriately completes the request.
  –Instruction:
  Introduce movie Raising Arizona (1987) and describe its attributes (including but not limited to genre, director/cast, country, character, plot/theme, mood/tone, critical acclaim/award, production quality, and soundtrack).
  –Response:
- **Answer:**
  Raising Arizona is a comedy film directed by the Coen brothers and released in 1987. The main cast includes Nicolas Cage, Holly Hunter, and John Goodman. The film was produced in the United States.
  The plot of Raising Arizona is centered around a couple, played by Cage and Hunter, who are unable to have children of their own. They decide to kidnap one of a wealthy family's quintuplets, but soon face trouble as the child's biological parents hire a bounty hunter to track them down.
  The movie's tone is quirky and absurdist, with elements of slapstick comedy and witty dialogue. The soundtrack features a mix of country music and traditional film score.
  Raising Arizona received critical acclaim upon its release and has since become a cult classic. The film's unique blend

of humor, character development, and visual style has made it a beloved favorite among fans of the Coen brothers' work. The movie was nominated for several awards, including a Golden Globe for Best Performance by an Actor in a Motion Picture - Musical or Comedy for Nicolas Cage's performance.

Product Relation Prediction.

- **Instruction:**
  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
  –Instruction:
  Given the title of two products, predict if the two products are similar, if the two products will be purchased or viewed together. Answer only from the options.
  –Input:
  Product 1:: Gamo Whisper Fusion Pro 0.22-Calibre Air Rifle with 3-9x40 Adjustable Objective Scope,
  Product 2:: RWS Model 34 .22 Caliber Pellet Air Rifle Combo
  –Options:
  A: Users who buy product 1 may also buy product 2.
  B: Users who view product 1 may also view product 2.
  C: The product 1 is similar with the product 2.
  –Response:
- **Answer:**
  "B"

Product Matching.

- **Instruction:**
  Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
  –Instruction:
  Given the title, description, manufacturer, and price of two products, identify if they are the same product. Only output yes or no.
  –Input:
  "product 1": {"title": "spycatcher 2007for pc", "description": NaN, "manufacturer": "avanquest", "price": "19.99"},
  "product 2": {"title": "avanquest usa llc spycatcher 2007", "description": "detect and remove spyware spycatcher is the first and only anti-spyware solution that continuously protects consumers from next-generation mutating spyware easily and safely. spycatcher enables even novice pc users to remove and block the most ins", "manufacturer": "nan", "price": "17.5"}
  –Response:
- **Answer:**
  yes

User2Item.

- **Query:**
  Generate a representation of this user sequence to find the next product this user will buy (User sequence is in reverse chronological order):
  Rawlings Players Series Youth T-Ball Glove, Regular, Basket-Web, 9 Inch
  ProsourceFit Flex Foam Rollers

WeeRide Kangaroo Child Bike Seat
FOCO NHL Unisex Team Thematic Gnome

- **Positive Sample:**
  Represent this product to match a user sequence:
  Start Smart Kids Fun Silicone Swim Cap for Boys and Girls - Sharks & Minnows.
- **Negative Sample:**
  Encode this product to fit a user sequence:
  Gamo Rocket .177 Cal, 9.6 Grains, Ballistic Tip, 150ct.

Query2Item.

- **Query:**
  Create an embedding of this user query to retrieve a relevant item:
  floral napkins baby shower
- **Positive Sample:**
  Utilize the product's title to form a representation that corresponds to a user query:
  Vintage Floral Paper Napkins for Bridal Shower, Tea Party & Luncheon (6.5 x 6.5 In, 100 Pack)
- **Negative Sample:**
  Represent the given product with a title to match a user query:
  AIMEILI Nail Dipping Powder Nail Art Powder 6 Colors for Dipping Nail

Item2Item.

- **Query:**
  Using the product's title and categories, produce an embedding to retrieve items that are alike:
  title: Terra Men's Pilot, categories: Fashion
- **Positive Sample:**
  Using the attributes and details of an item, create an embedding to help find similar products:
  title: Terra Men's Pilot, categories: Fashion
- **Negative Sample:**
  Based on an item's details and attributes, produce an embedding to identify similar products:
  title: Crew Socks For Men Women Glamour Hot Pink Glitter Cotton Performance Cushion Sport Athletic With Moisture Wicking, categories: Fashion