Leveraging Sentence-T5 for Sequential Recommendation

Sahil Jethani

Master of Science
School of Informatics
University of Edinburgh
2024

Abstract

This research explores the application of large language models (LLMs) for sequential recommendation tasks. We propose a novel approach that reformulates sequential recommendation as a sentence retrieval problem, leveraging the Sentence-T5 (ST5) model, an LLM specifically pretrained to generate high-quality sentence embeddings for various sentence-level tasks. Through extensive experimentation, we developed an effective method for converting user sequence histories and item descriptions into sentences, which are then encoded into sentence embeddings using our enhanced ST5 model fine-tuned for this task.

In our study, we identify two key challenges in utilizing the pretrained ST5 model: the semantic gap and limited sequence awareness. To address these issues, we developed a novel two-phase pretraining approach. First, we employ Item-Description contrastive learning to bridge the semantic gap. Second, we implement Sequence-Sequence contrastive learning to enhance sequence awareness. Following these pretraining phases, we fine-tune the model using Sequence-Item contrastive learning. This comprehensive approach results in our enhanced ST5-Final model, which demonstrates significant improvements over strong baselines such as SASRec and UniSRec (BLaIR) across nine diverse product categories from the Amazon Reviews'23 dataset.

Our ST5-Final model not only performs well on trained categories but also on unseen product categories and non-e-commerce platforms. By demonstrating our model's effectiveness in rating prediction, we prove its ability to generate universal item and user representations applicable to various recommendation tasks. This universality across domains, platforms, and recommendation tasks suggests that our work may contribute towards the development of a foundation model for recommendation systems.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Sahil Jethani*)

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Timos Korres, Phong Le, and Vladimir Eremichev, for their constant support and guidance throughout this dissertation. I am deeply thankful to The University of Edinburgh and Professor Chris Williams for providing me the opportunity to work on this master's project in collaboration with Amazon, and for the crucial GPU resources that made this research possible.

A huge thanks to my friends Akshath, Aditya, Thejus, and Vaikunth for your constant support, all those all-nighters, and for keeping me sane through this intense journey.

Finally, a heartfelt thanks to my family and Lea. Your unconditional love and belief in me have been my anchor. Through the sleepless nights, the moments of self-doubt, and the small victories, you've been there, cheering me on and supporting me. Thank you for being my constant in this whirlwind of a year.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

In modern times, recommendation systems have become a crucial part of our online experiences. These systems guide users through a vast amount of options, from suggesting products on e-commerce platforms [16] to recommending content on streaming services [38]. As users engage with digital platforms over time, their preferences evolve, which creates a need for more sophisticated recommendation approaches that can capture these dynamic patterns[3].

To address the evolving nature of user preferences, sequential recommendation has emerged as a promising approach. In contrast to traditional recommendation methods that treat user preferences as static, sequential recommendation models approach the task from a dynamic perspective [51]. The primary objective of sequential recommendation is to predict the next item a user is likely to interact with, based on their historical sequence of item interactions.

The domain of sequential recommendation has experienced significant growth in recent years, mirroring the progress in the natural language processing (NLP) field [53]. This parallel evolution is not surprising, as sequential recommendation can also be viewed as an NLP task, with items similar to words and user sequences comparable to sentences. Early methods utilized Recurrent Neural Networks (RNNs) to model sequential data, as demonstrated by Hidasi et al.[15] with their GRU4Rec model. The introduction of Transformer architectures [49] marked a significant leap forward, with models such as SASRec [22], BERT4Rec [43], S3-Rec [67], and CL4SRec [55] enabling more effective modeling of long-range dependencies in user behavior sequences.

Inspired by the powerful performance of LLMs [4] across numerous domains [36],

researchers have begun exploring their potential for sequential recommendation systems [66]. This exploration involves converting the sequential recommendation problem into a textual format, offering advantages such as leveraging pre-trained knowledge and addressing limitations of ID-based methods [30].

To effectively utilize LLMs for sequential recommendation tasks, several challenges need to be addressed. It remains unclear how to formulate the sequential recommendation problem as a textual task. Additionally, the mismatch between LLMs' objective to solve natural language understanding tasks and the goals of recommendation systems creates a semantic gap [30, 35, 17]. Furthermore, LLMs exhibit poor user sequence modeling capabilities in recommendation settings [18], a crucial limitation for effectively leveraging long user histories in sequential recommendation tasks.

Moreover, there is a need to develop universal representations for users and items since they can facilitate cross-domain and cross-platform recommendations without frequent retraining [17]. Such representations can also support various recommendation tasks beyond sequential recommendation, such as rating prediction.

1.2 Our Approach and Contributions

Our approach reformulates sequential recommendation as a sentence retrieval task, leveraging LLMs. We convert user sequences and item descriptions into sentences, then utilize the Sentence-T5 base model (ST5) [32] to encode them into sentence embeddings. We then match user sequence embeddings to the item corpus to retrieve recommended items. ST5 is specifically designed for encoding sentences into high-quality embeddings suitable for various sentence-level tasks, including retrieval. Through careful experimentation, we first develop an effective method to convert user sequences and items into textual descriptions. We then identify and address two primary limitations of using ST5 for sequential recommendation: the semantic gap and limited sequence awareness.

To overcome these challenges, we develop a novel two-phase pretraining approach. First, we pretrain our model on an Item-Description contrastive learning, which utilizes item descriptions to address the semantic gap between sentence retrieval for natural language understanding tasks and item retrieval in sequential recommendation settings. We then conduct pretraining on a Sequence-to-Sequence contrastive learning, inspired by CL4SRec [55], to enhance the model's sequence awareness—crucial for capturing temporal dynamics in user interactions. After pretraining, we fine-tune on a Sequence-

Item contrastive learning objective for our core task of sequential recommendation, training the model to retrieve top items for a user.

To ensure our model generates universal item and user representations, we train it simultaneously on diverse product catalogs from nine different domains in the Amazon Reviews'23 dataset [16]. We demonstrate the model's cross-domain capabilities on three unseen product categories and validate its cross-platform applicability on the Yelp (2018) dataset [60]. Additionally, we apply our model to rating prediction to prove its universality for recommendation tasks beyond sequential recommendation.

Our key contributions can be summarized as follows:

1. We develop an effective method for converting sequential recommendation task into a text-based format.

2. We identify the limitations of the ST5 model for this task.

3. We enhance the ST5 model for sequential recommendation task by addressing its inherent limitations through tailored pretraining and fine-tuning strategies.

4. We demonstrate that our proposed and trained ST5 model is capable of generating universal user and item representations with strong generalization capabilities across diverse domains, platforms, and other recommendation tasks.

1.3 Structure

This research provides a comprehensive exploration of our novel approach to sequential recommendation. We begin with background on sequential recommendation and the Sentence-T5 (ST5) model in Chapter 2, followed by our methodology in Chapter 3. Chapter 4 forms the core of our work, discussing the conversion of the sequential recommendation problem into a sentence retrieval task, analyzing the ST5's zero-shot performance, and detailing our training process. Chapter 5 explores various experiments, focusing on cross-domain applicability and effectiveness in different tasks. We conclude in Chapter 6 with a discussion of our model's limitations, future work, and final remarks.

Chapter 2

Background

In this chapter, we explore the fundamentals of sequential recommendation systems and their types. We examine related research to position our work within the field. We then introduce the Sentence-T5 (ST5) model, our chosen sentence encoder, and explain why it was an ideal choice for our project.

2.1 Sequential Recommendation

Recommendation systems have become integral to digital experiences, enhancing both business operations and user experiences. They help businesses drive engagement and sales while providing users with personalized suggestions [3]. Traditional approaches to recommendation systems fall into two main categories. Collaborative filtering [42] analyzes patterns of user behavior across a large user base to make recommendations based on similar preferences. Content-based recommendation [46] creates profiles for users and items, recommending items similar to those a user has liked in the past.

Unlike traditional methods which treat user preferences as static, sequential recommendation systems capture the dynamic nature of user interests and item spaces by analyzing the user's interactions with different items under varying contexts [51]. These systems aim to capture temporal dependencies in user interaction patterns, understanding both short-term and long-term evolving implicit preferences. This not only improves the user experience by offering more personalized recommendations but also benefits businesses by increasing engagement and potentially driving sales through more targeted suggestions [3].

Sequential recommendation systems can be categorized based on the types of behavior sequences they analyze as explained by Fang et al.[8]. Experience-based

sequences capture multiple interactions with the same item through different behaviors such as clicking, purchasing, or sharing. Transaction-based sequences concentrate on interactions involving a single type of behavior, most commonly purchases. Interaction-based sequences combine aspects of both experience-based and transaction-based sequences. Our research will concentrate on transaction-based behavior sequences, which align well with popular datasets like Amazon Reviews'23 [16].

2.2 Type of Sequential Recommendation

Sequential recommendation systems can be broadly categorized into mainly two types, ID-based and textual-based approaches. ID-based sequential recommendation systems have been the mainstream approach for a long time, relying on unique identifiers for users and items to generate personalized recommendations. This approach has evolved from the traditional methods such as Markov chain-based methods [40] to more advanced techniques using transformer architectures [22]. However, this approach faces limitations such as dependency on sufficient user-item interaction data, difficulty in capturing attribute-level correlations that reflect real user preferences, and challenges in cross-domain capabilities [30]. In contrast, textual-based approaches have emerged as a promising alternative, leveraging rich textual information to represent items and/or users without explicitly involving IDs [25, 35]. In the following sections, we will explore these two approaches in greater detail:

2.2.1 ID-Based Sequential Recommendation Models

Traditional methods laid the foundation of the ID-based systems, with Markov chain-based approaches [68] and matrix factorization techniques [23] being among the first to capture temporal aspects of user behavior. Major advancement came with Rendle et al.'s [40] Factorized Personalized Markov Chains (FPMC) model, which combined the Markov chain models with matrix factorization to do next-basket recommendations in an e-commerce setting. However, this approach faced several limitations, such as struggles with long-term dependencies and complex patterns [51].

Recurrent Neural Networks (RNNs) emerged as a promising solution, with models such as GRU4Rec [15] that effectively captured longer sequences and more complex sequence dynamics. The use of Convolutional Neural Networks (CNNs) [24] and Graph Neural Networks (GNNs) [41] followed next in the field, with models like Caser [44]

and SR-GNN [54], which offered a more flexible framework and improved capability in capturing local patterns for modeling user behavior.

The introduction of transformer-based models [49] marked a revolution in the field of sequential recommendation. These models had the combined ability to capture long-range dependencies and more efficient parallel processing, overcoming the limitations of the previous approaches. SASRec [22] adapted the Transformer architecture for sequential recommendation, using a two-layer Transformer and self-attention mechanism to better model the user sequences. BERT4Rec [43] further advanced this approach by introducing bidirectional self-attention, allowing the model to consider both past and future interactions. The bidirectional architecture coupled with a masked item prediction objective allowed the model to generate more context-aware representations of the user behaviors. Self-supervised approaches led the next step by using self-supervision signals to generate better data representations and address data sparsity issues. S3-Rec [67] introduced a self-supervised learning framework that pre-trains the model on large-scale unlabeled sequences, incorporating item, attribute, and position information. Meanwhile, CL4SRec [55] used contrastive learning techniques to improve the robustness of user sequence representations by generating augmented versions of user sequences through operations like item crop, item mask, and item shuffle. These techniques further improved the transformer-based models and represent the current state-of-the-art in ID-based sequential recommendation systems.

2.2.2 Text Based Sequential Recommendation Models

Text-based sequential recommendation models represent a paradigm shift in the field. Leveraging rich textual information to represent items and user interactions, these models improve handling of cold-start scenarios, enhance cross-domain capabilities, and show promise for more explainable recommendations [10]. Researchers are exploring various ways to use Large Language Models (LLMs) [4] in recommendation systems, given their significant success across numerous natural language processing tasks [36].

One approach uses in-context learning and prompt engineering, which involves crafting specific prompts to guide LLMs in solving recommendation tasks without fine-tuning. Gao et al. introduced Chat-REC [9], an LLM-augmented recommender system that uses in-context learning to enhance recommendation reasoning. Wang and Lim [50] proposed a Zero-Shot Next-Item Recommendation (NIR) prompting strategy for next-item predictions. Hou et al. [18] reformulated sequential recommendation as a con-

ditional ranking task, demonstrating the potential of LLMs as zero-shot rankers. While promising, these approaches often struggle with perceiving user sequence interaction order and can have a popularity bias in recommending items.

Another line of research leverages LLMs by framing recommendation as an instruction-tuning problem. This approach aims to train LLMs as one-model-fits-all solutions by unifying various recommendation tasks under a single model. The M6 model [5] introduced this concept for open-ended domains and tasks in industrial recommender systems. The P5 model [10] developed prompt templates for various recommendation tasks and fine-tuned the T5 model. VIP5 [11] built on P5 by incorporating images alongside text. InstrucRec [61] formulated user preferences as natural language instructions, treating recommendation as an instruction-following task for LLMs. Our research specializes in sequential recommendation rather than attempting to cover all recommendation tasks.

2.3 Relevant Work

Our research aligns with the text-based sequential recommendation approach, specifically focusing on using LLMs as powerful text encoders to generate user and item representations [63]. Several recent works have explored similar directions. IDA-SR (Item Description-based Sequential Recommendation) [30] used a BERT model [7] to generate item representations from textual descriptions. It then applies multi-head attention to a sequence of these items from user interaction histories to create user sequence representations. UniSRec [17] introduced the concept of universal item and sequence representations to further improve cross-domain capabilities and reduce ID dependence. The BLaIR (Bridging Language and Items for Retrieval and Recommendation) model [16] further refines item representations by introducing Item-Review contrastive learning, aiming to bridge the semantic gap between natural text and item text, and then uses UniSRec as a backbone for sequential recommendation. Our approach diverges from these methods by converting both item descriptions and user sequences into textual format to fully leverage the LLMs. We address the semantic gap through pretraining objectives for both sequence and item representations in a single model, introducing Item-Description pretraining instead of BLaIR's Item-Review contrastive learning.

The Unified Pre-trained Language Model Enhanced Sequential Recommendation (UPSR) [35] uses a T5 encoder-decoder [37] model as its backbone, converting user sequences into text and framing sequential recommendation as an item generation task.

Recformer [25] implements an approach similar to ours, using sentence retrieval for sequential recommendation. It employs a Longformer-like model [2] as its backbone and obtains both user and item representations from the same model. Our approach differs from Recformer in two key aspects. First, we utilize the Sentence-T5 (ST5) model [32], which is specifically trained for sentence retrieval tasks, thus providing a more suitable foundation for recommendation tasks. Second, our structured input method clearly differentiates between sequence and item representations, explicitly instructing the model when to produce each type. Liu et al. [27] also formulate sequential recommendation as a sentence retrieval task using a T5 model as the backbone. However, our model goes further by addressing the inherent weaknesses of using an LLM like ST5 for recommendation task. Furthermore, our approach focuses on creating universal representations for both items and users, which not only enhances performance across diverse domains but also enables the application of our representations to other downstream recommendation tasks, such as rating prediction.

2.4 Sentence-T5 (ST5) Model

In our research, the selection of an appropriate sentence encoder is crucial for effectively transforming our sequential recommendation task into a sentence retrieval problem. After careful consideration and preliminary experiments, we chose the Sentence-T5 (ST5) [32] model as our primary sentence encoder, driven by several key factors aligning with our task requirements and broader research goals.

The ST5 model, which utilises T5's encoder architecture as its backbone [37], excels in generating high-quality sentence embeddings suitable for various sentence-level tasks, including sentence retrieval tasks. In the context of recommendation systems, which operate on large datasets of users and items, efficiency is crucial. Using a standard T5 model for sentence retrieval tasks would have involved computationally expensive cross-attention on each query-candidate pair. Instead, leveraging sentence embeddings proves to be a more efficient approach [12, 39, 59].

ST5's promising results, even without task-specific fine-tuning, provide an excellent starting point for our sequential recommendation task. This aligns well with our research goals of developing universal item and user representations that can generalise across domains and platforms. The ST5's strong foundation in sentence retrieval and semantic search and the flexibility for further fine-tuning to our specific task make it an ideal choice for our research.

Chapter 3

Methodology

Chapter 3 outlines our research methodology. We define our approach to sequential recommendation, evaluation metrics,datasets and preprocessing steps, and provide an overview of the baseline models used for comparison in our study.

3.1 Task Formulation

The sequential recommendation task aims to predict a user's next item of interest based on their historical behavior. Given a set of items I and a list of user sequences U, where each user sequence $s \in U$ comprises a list of items $i_t \in I$ that the user interacted with at time t, such that $s = (i_1, i_2, i_3, \ldots, i_t)$, our objective is to predict the item at i_{t+1}. Specifically, we aim to retrieve the top K items that users are most likely to interact with at timestamp i_{t+1}, where K is the number of recommended items. For our research, we select K to be 10 and 50. The choice of K = 10 helps us simulate common real-world scenarios with limited recommendation space, while K = 50 allows us to evaluate the model's performance over a broader range of recommendations and assess the model's ability to capture diverse user interests.

We reformulate this challenge as a sentence retrieval task, converting each item and user sequence into a sentence with an appropriate structure as detailed in Section 4.1. To encode these sentences into high-dimensional sentence embeddings, we leverage the Sentence-T5 base (ST5) [67] as our backbone LLM. Let the sentence encoder be denoted by the function $f()$. We obtain embeddings for items and sequences as follows:

$$\text{Emb}_i = f(\text{sent}_i) \in R^d; \quad \text{Emb}_s = f(\text{sent}_s) \in R^d \tag{3.1}$$

where d is the embedding dimension which for ST5 is 768, and sent_i and sent_s

represent the textual sentences for items and sequences respectively. We calculate the similarity between a user sequence and all items using cosine similarity:

$$\text{sim}(\text{Emb}_s, \text{Emb}_i) = \frac{\text{Emb}_s \cdot \text{Emb}_i}{|\text{Emb}_s||\text{Emb}_i|} \tag{3.2}$$

To improve efficiency and scalability, we adopt an approach similar to [58]. We calculate and store the item corpus representation matrix in advance, while computing user sequence representations in real-time as needed.

The transformation of this task into a sentence retrieval problem offers several advantages. Firstly, it removes the dependency on item IDs, a limitation faced by many existing models [40]. Secondly, the text-based approach allows for better bridging of cross-domain and cold-start scenarios [17]. Furthermore, by utilizing LLMs, we can leverage their expressive power, pre-trained knowledge, and semantic capabilities to provide more personalized, context-aware recommendations [35].

Our methodology involves designing pretraining tasks aligned to address the current limitations of ST5. The fine-tuning process is then tailored to optimize performance for the sequential recommendation task. This two-step approach leverages the broad knowledge in pretrained ST5 while adapting it to generate universal item and user representations applicable across domains, platforms, and other recommendation tasks. These training steps are later described in Chapter 4.

3.2 Evaluation Metrics

To assess the performance of our model, we employ two widely used metrics in recommendation systems: Hit Ratio (HR@K) [6] and Normalized Discounted Cumulative Gain (NDCG@K) [20]. These metrics provide insights into the effectiveness of our recommendation model.

The Hit Ratio (HR@K) measures the proportion of cases where the ground truth item is present in the top K recommended items. Let N be the total number of users, K the number of top items to consider, Top_k^n the set of top k predicted items for user n, and GT_n the ground truth item for user n. $\delta(\text{GT}_n \in \text{Top}_k^n)$ be an indicator function which equals to 1 if $\text{GT}_n \in \text{Top}_k^n$, and 0 otherwise. Then HR@K is calculated as:

$$\text{HR@K} = \frac{1}{N} \sum_{n=1}^{N} \delta(\text{GT}_n \in \text{Top}_k^n) \tag{3.3}$$

While HR@K is valuable, it doesn't account for the position of the ground truth item within the top K recommendations. To address this, we also use the NDCG@K, which

considers both the presence and the ranking of the ground truth item.The NDCG@K is calculated in several steps. First, we compute the Discounted Cumulative Gain (DCG@K) for each user n:

$$\text{DCG@K}_n = \sum_{i=1}^{K} \frac{\delta(\text{GT}_n = \text{item}_i)}{\log_2(i+1)} \tag{3.4}$$

Here, $\delta(\text{GT}_n = \text{item}_i)$ is 1 if the item at position i is the correct next item, and 0 otherwise. The logarithmic discount factor penalizes correct items appearing lower in the recommendation list.

Next, we calculate the Ideal DCG (IDCG@K), which represents the best possible ranking where the correct item appears at the top. In our case, $\text{IDCG@K}_n = 1$ for all users, as the correct next item would ideally be at the top of the list. Finally, we average the NDCG@K across all users:

$$\text{NDCG@K} = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{K} \frac{\delta(\text{GT}_n = \text{item}_i)}{\log_2(i+1)} \tag{3.5}$$

This metric not only considers whether the ground truth item is in the top K recommendations but also rewards higher placements within that list. NDCG is particularly relevant for e-commerce platforms, where the order of recommendations can significantly impact user engagement and sales.

3.3 Dataset

For training and evaluating our model, we utilize the Amazon Reviews'23 dataset, provided by McAuley Lab [16]. This comprehensive dataset, derived from Amazon [1], an e-commerce website, encompasses user-item interactions and detailed item metadata from 33 diverse product categories. It offers crucial information such as user IDs, item IDs, timestamps, ratings, and rich item metadata.

We selected nine diverse product categories: *All Beauty*, *Beauty and Personal Care*, *Cell Phones and Accessories*, *Electronics*, *Health and Household*, *Movies and TV*, *Toys and Games*, *Video Games*, and *Baby Products*. This wide-ranging selection enhances our model's ability to generalize across different domains and aids in developing universal representations. To rigorously evaluate cross-domain performance, we selected three distinct categories: *Books*, *Digital Music*, and *Amazon Fashion*. These categories present unique characteristics and challenges, testing our model's ability to transfer knowledge and make meaningful recommendations in diverse product spaces.

A primary motivation for selecting this dataset is its recency, covering user interactions from October 2018 to September 2023. This temporal range aligns with the knowledge cutoff of most LLMs, ensuring consistency between the training data and our backbone LLM model's knowledge. The enhanced item metadata provided in this dataset is another compelling factor. Rich textual item titles, descriptions, and features offer our model a deeper understanding of the items being recommended. This comprehensive textual information is particularly beneficial for our approach, as we rely on text to create item and user representations. The detailed item descriptions directly contribute to the effectiveness of our item and description pretraining objectives, as detailed in Section 5.2.1.

3.4 Dataset Preprocessing

Our data preprocessing for the Amazon Reviews'23 dataset involved several key steps to prepare the data for our sequential recommendation model. We adopted the Absolute-Timestamp Splitting strategy as specified by the dataset creators [16], dividing user interaction sequences based on specific timestamps: interactions before t_1 for training, between t_1 and t_2 for validation, and after t_2 for testing. This approach mirrors real-world scenarios where recommender systems can only utilize historical interactions up to a certain point in time.

We cleaned the item metadata, focussing on item titles and descriptions, by converting HTML entities, removing HTML tags and non-ASCII characters, and normalising whitespace. The dataset offers k-core filtering options, where k-core retains only users and items with at least k interactions. We opted for 0-core filtering, which keeps all users and items regardless of interaction frequency. This approach helps maintain comprehensive data, maximize diversity, and preserve cold-start scenarios. To manage computational constraints, we selected subsets from each product category using a 7:2:1 ratio for training, validation, and testing. Data statistics are given in Appendix C.

Throughout the preprocessing, we removed items without metadata from the user-item interaction data and retained only histories with at least one item. A key challenge was accommodating the ST5 model's input length limitation of 255 tokens. For user interaction histories, we prioritized recent interactions by removing items from the beginning of the sequence until the token limit was met. For item titles, we truncated from the end while meticulously preserving all necessary tags, maintaining the integrity of both item and sequence representations. Finally, we created data maps for users,

items, titles, and descriptions, facilitating efficient data retrieval and model processing. This indexing step is crucial for managing the large-scale dataset and enabling quick lookups during model training and evaluation.

3.5 Baseline Models

To rigorously evaluate the effectiveness of our proposed model, we compare it against three well-established baseline models. Each of these baselines represents a different approach to the recommendation task, providing a comprehensive framework for assessing the strengths and weaknesses of our model in various scenarios.

3.5.1 Popularity-based Model (Pop)

This baseline identifies the most frequently interacted items in the training data, ranks them by popularity, and recommends this fixed ranking to all users in the test set. Despite its simplicity, it can be surprisingly effective, particularly in entertainment categories like *Video Games*, *Movies and TV*, or *Toys and Games*, where trending items often drive consumer behavior. It's also potentially strong in domains like *Cell Phones and Accessories* or *Electronics*, where consumers often gravitate towards the most popular models. By comparing our ST5-based model against this method, we can assess its ability to capture personalized preferences beyond general popularity trends.

3.5.2 Self-Attentive Sequential Recommendation (SASRec)

SASRec (Self-Attentive Sequential Recommendation) is a state-of-the-art model for sequential recommendation tasks [22]. We choose SASRec as a baseline due to its strong performance in capturing complex user behavior patterns through self-attention mechanisms. In SASRec, a user's sequence $s \in U$ is converted into embeddings using an item embedding matrix $M \in R^{I \times d}$, where I is the total number of items and d is the embedding dimension. The input embedding matrix $E \in R^{n \times d}$ is constructed by mapping each item i_t at time step t to its corresponding embedding $E_t = M_{s_t}$, where n is the sequence length.

The model then applies self-attention blocks to these embeddings to capture dependencies between items. The self-attention mechanism [49] computes the attention

weights and the context-aware representations using the formula:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{3.6}$$

where $Q = EW^Q$, $K = EW^K$, and $V = EW^V$ are the query, key, and value matrices derived from the item embeddings E, and W^Q, W^K, and W^V are learnable parameter matrices. These attention weights determine the relevance of each item in the sequence and generate context-aware item representations. For the final prediction, these context-aware representations are passed through feed-forward layers. The model predicts the next item by computing a relevance score for each potential item using:

$$r_{i,t} = F_t \cdot M_i^T \tag{3.7}$$

where F_t is the final sequence representation after the self-attention blocks, and M_i is the embedding of item i. The item with the highest score is predicted as the next item.

SASRec is an ID-based method that relies on learned item embeddings. In contrast, our method uses the ST5 encoder to create text-based representations of items and user sequences. By comparing SASRec with our approach, we aim to evaluate the trade-offs between ID-based and text-based sequential recommendation approaches, particularly in terms of performance across different domains and cold-start performance.

3.5.3 Universal Sequence Representation Learning (UniSRec(BLaIR))

In our study, we use UniSRec(BLaIR) as a strong comparative baseline. This model combines the Universal Sequence Representation Learning (UniSRec) framework [17] with item representations derived from the BLaIR model [16].This combination leverages UniSRec's ability to perform cross-domain sequential recommendations while harnessing BLaIR's sophisticated item embeddings.

UniSRec aims to generate universal item and sequence representations capable of generalizing across various domains. It takes a user's interaction sequence as input, converting each item in the sequence to its textual description. To create item representations, the text of each item is processed through a pre-trained BERT model [7] to obtain initial sentence embeddings. These embeddings are then processed through a parametric whitening step and a Mixture-of-Experts (MoE) enhanced adaptor to create universal item representations v_i. This step ensures that the embeddings are evenly distributed across the latent space and adaptable across domains, improving the model's ability to generalize.

The sequence of universal item representations v_i is then encoded using a Transformer-based architecture. The input to this encoder is : $f_j^0 = v_i + p_j$, where p_j are positional embeddings. The Transformer applies multiple layers of self-attention and feed-forward networks:

$$F^{(l+1)} = \text{FFN}(\text{MHAttn}(F^l)) \tag{3.8}$$

where F^l is the output of the l-th layer, MHAttn denotes multi-head self-attention, and FFN represents a point-wise feed-forward network. After L layers, the final hidden state f_n^L corresponding to the n-th (last) position is used as the sequence representation. For next item prediction, the model's final output is a probability distribution over the entire item catalog, predicting the next item a user is likely to interact with. UniSRec computes this probability for each candidate item j as:

$$P(j|s) = \text{Softmax}(f_n^L \cdot v_j) \tag{3.9}$$

where v_j is the universal representation of item j.

BLaIR is a language model designed to create universal item representations. It employs an Item-Review contrastive learning objective to bridge the semantic gap between natural language and item representations for retrieval and recommendation tasks. The resulting item embeddings are then utilized in UniSRec, leveraging these rich, contextually aware representations for more effective recommendations.

We selected UniSRec(BLaIR) as a baseline due to its strong performance in cross-domain recommendation tasks. While UniSRec(BLaIR) uses text-based representations similar to our approach, we further innovate by representing both items and user behaviors as text and utilizing Item-Description contrastive learning instead of Item-Review contrastive learning (explained in Section 4.4.1.1). This comparison will offer valuable insights into the effectiveness of our approach in creating universal representations for both users and items in recommendation tasks.

3.6 Implementation Details

This project was conducted on the Eddie cluster, utilizing A100 GPUs [45]. We implemented SASRec and UniSrec(BLaIR) baselines using the RecBole library [65, 56, 64]. Our ST5 model and its variants were developed and trained using the sentence-transformers library [39]. Detailed model configurations are explained in Appendix B.

Chapter 4

ST5 Model Development

This chapter outlines the development of our sequential recommendation model from initial text representation experiments to the final optimized Sentence-T5 (ST5) model [32]. We begin by exploring strategies to effectively transform the sequential recommendation task into a sentence retrieval problem, followed by an analysis of ST5 zero-shot performance, which we referred to as "ST5-Only." After identifying key weaknesses, we detail the development of our enhanced model, "ST5-Final," which incorporates specialized pretraining and fine-tuning phases.

4.1 Text Representation Experiment

In our research on sequential recommendation, a critical aspect was determining the most effective strategy for transforming item information and user sequences into a textual format suitable for sentence retrieval tasks. This transformation process is crucial as it directly impacts the quality of sentence embeddings and, consequently, the performance of our sequential recommendation system. To identify the optimal approach, we conducted a series of experiments focusing on different text representation strategies.

4.1.1 User Sequence Text Representation

Our initial experiment focused on evaluating various methods for modeling user sequence into sentences , exploring four distinct approaches: Structured, Unstructured, Structured with Instruction, and Unstructured with Instruction.

The Structured Approach encapsulated item information within specific tags, using

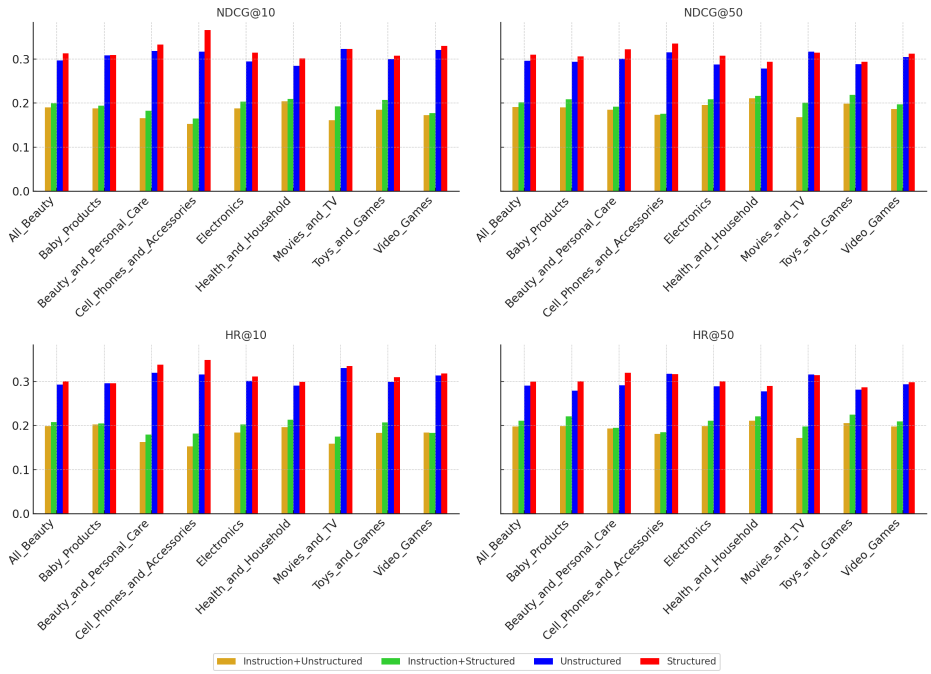

Figure 4.1: Comparison of User Sequence Text Representation. The top two charts show NDCG@10 and NDCG@50 scores, while the bottom two show HR@10 and HR@50 scores. The bars are normalized to show the proportion of each strategy's contribution within each category.

only the item title as the item text. The entire user sequence was bounded by start and end tags. We utilized T5 additional tokens to represent these special markers as follows: <SEQ_START> as <extra_id_0>, <ITEM_START> as <extra_id_1>, <ITEM_END> as <extra_id_2>, and <SEQ_END> as <extra_id_3>. This format was designed to help the model distinguish between individual items and potentially capture long-term dependencies in user behavior. Additionally, these tags assist the model in differentiating when to create sequence embeddings and when to generate item embeddings. In contrast, the Unstructured Approach represented items simply as their text, with the sequence being a comma-separated list of items.

The Structured with Instruction and Unstructured with Instruction approaches combined their respective formats with an introductory instruction:

> A user has purchased a sequence of items ordered in chronological order. Each item in the sequence is represented as "Title: <item title>". The following sentence represents the user history:

This instruction was followed by either the structured or unstructured sequence

representation.

Our results, as presented in Figure 4.1 (Table in Appendix A.3), demonstrated a clear hierarchy of effectiveness. The structured approach consistently outperformed the others, followed by the unstructured approach. The addition of instructions generally reduced performance, likely due to the introduction of noise in the sentence representation process. However, when instructions were used, the structured format still outperformed its unstructured counterpart. These findings suggest that providing clear item boundaries and sequence structure helps the model better understand the relationships between items and the overall user behavior pattern. The superior performance of the structured approach without instructions indicates that the model can effectively leverage the inherent structure without additional explanatory text.

4.1.2 Item Text Representation

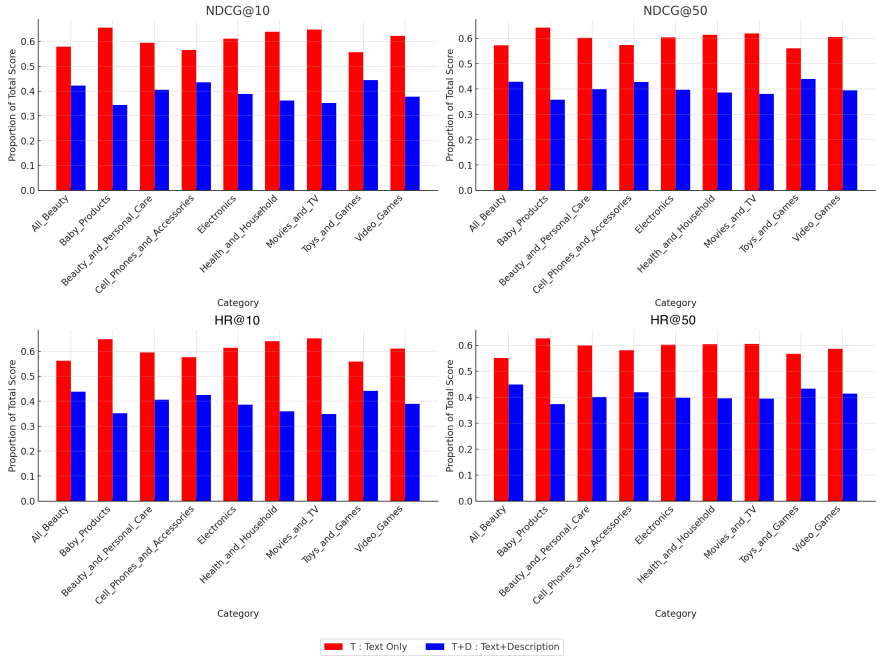

Figure 4.2: Comparison of Item Text Representation. The top two charts show NDCG@10 and NDCG@50 scores, while the bottom two show HR@10 and HR@50 scores.The bars are normalized to show the proportion of each strategy's contribution within each category.

Our second experiment delved into the finding out the an effective way to represent item text. We explored two distinct strategies: using only the item title, and combining the title with its description. For the title-only approach, we formatted the data as:

<ITEM_START>TITLE: <ITEM TITLE><ITEM_END>

In contrast, the combined method used the following format:

<ITEM_START>TITLE: <ITEM TITLE>DESCRIPTION: <ITEM DESCRIPTION><ITEM_END>

If an item lacked a description, we used the placeholder "description not available." Surprisingly, our results, as illustrated in Figure 4.2 (Table in Appendix A.4), revealed that using the title alone yielded superior outcomes compared to including the description. This counterintuitive outcome can be attributed to several key factors. Primarily, item titles, especially in e-commerce contexts, usually pack a lot of important details about the product, such as brand, shape, features, colour, etc. This concentrated information allows for a comprehensive representation of the item. Additionally, by focusing solely on the title, we potentially mitigate the noise inherent in product descriptions, which often contain marketing language or redundant information that may not significantly contribute to the item's core representation. Furthermore, the concise and targeted nature of titles may result in more distinct and easily distinguishable embeddings in the vector space. This could potentially enhance the model's capacity to differentiate between items.

4.2 ST5-Only Model Performance Analysis

As observed in Figure 4.3 (Table in Appendix A.1), our evaluation of the ST5-Only model against baselines reveals a consistent performance ranking across product categories: UniSREC (BLaIR) generally leads, followed by ST5-Only, then SASRec, and finally the Popularity (Pop) base method. Notably, in the *All Beauty* category, ST5-Only even surpasses UniSRec (BLaIR). These results highlight our ST5's zero-shot potential, offering strong competition to sophisticated, domain-adapted models.

Our sentence retrieval approach demonstrates significant potential, often outperforming SASRec across various categories. Unlike ID-based methods, our approach's reliance on textual representations offers inherent cross-domain capabilities. However, the ST5-Only model's performance is not uniform across all categories. We observe certain challenges in domains such as *Cell Phones and Accessories* and *Electronics*. This variability likely stems from a semantic gap between the textual representation of

items and the specific requirements of recommendation tasks in these categories. The technical nature of these products may not align as closely with the language model's pretraining, which focuses on general semantics of the natural language understanding task. Conversely, the strong performance of ST5-Only in categories like *Movies and TV* can be attributed to the model's extensive world knowledge, acquired through pretraining on vast amounts of diverse textual data.

UniSRec (BLaIR) shows consistently strong performance across all metrics. This is due to its use of an LLM that was further trained with an item-review contrastive objective. This approach helps UniSRec bridge the semantic gap that the ST5-Only model struggles to overcome. Addressing this semantic gap could further enhance ST5's effectiveness across a broader spectrum of domains.

4.3 Weakness of ST5-Only Model

Following our analysis of the ST5-Only model's performance, it is crucial to address the limitations that have become apparent through our study. These weaknesses not only provide insight into the model's current capabilities but also highlight areas for future improvement.

1. **Semantic Gap in Recommendation Tasks:** As discussed in Section 4.2, a significant semantic gap exists between the ST5 model's pre-training for retrieving sentences based on natural language understanding and its application to retrieve items relevant to user sequences in a sequential recommendation context. This gap is particularly noticeable in the model's struggle to capture item-specific attributes and relationships essential for effective item retrieval. While models such as BLaIR [16] bridge this gap through targeted training on Item-Review contrastive objective, the zero-shot ST5 model lacks the specialized understanding of item semantics in the context of recommendations. Furthermore, this gap is evident in the model's uneven world knowledge across different product categories, with the model excelling in domains like *Movies and TV* and *Beauty products* but struggling in more technical categories such as *Electronics*.

2. **Limited Sequence Awareness:** Analysis of ST5-Only's performance with increasing user interaction history (Section 5.2.6) reveals a critical limitation in the model's ability to leverage sequential information effectively. The model's performance consistently declines as the number of interacted items in user history

increases, with optimal results observed for single-item sequences. This pattern suggests that the ST5-Only model primarily engages in sentence matching rather than demonstrating an understanding of user sequential patterns in a sequential recommendation context.

4.4 ST5-Final Model Development

Building upon the insights gained from our analysis of ST5-Only, we developed ST5-Final model to address the identified limitations and enhance the model's performance in sequential recommendation tasks. ST5-Final incorporates two key pretraining phases designed to bridge the semantic gap and improve sequence awareness, followed by a task-specific fine-tuning phase.

4.4.1 Pretraining

4.4.1.1 Item-Description Contrastive Pretraining

To address the semantic gap weakness identified earlier, we developed a novel pretraining approach that contrasts items with their corresponding descriptions. To the best of our knowledge, we are the first to apply this specific pretraining method in this context. This method trains the model to focus on relevant linguistic features when creating item representations, bridging the gap between natural language text for sentence retrieval and item text for retrieving relevant items for a user with a given user sequence. This approach is particularly beneficial for domains such as *Movies and TV* and *Video Games*, where item titles often consist of just the movie or game name, and descriptions provide crucial context. For example, consider an item title and its description from the *Movies and TV* category of the Amazon Reviews '23 dataset [16]:

<ITEM_START>TITLE: Big Hero 6 (Blu-ray+DVD+Digital HD) <ITEM_END>

<ITEM_START>DESCRIPTION: With all the heart and humor audiences expect from Walt Disney Animation Studios, BIG HERO 6 is an action-packed comedy adventure that introduces Baymax, a lovable, personal companion robot, who forms a special bond with robotics prodigy Hiro Hamada.Bring home Disney's BIG HERO 6, featuring comic-book-style action and hilarious, unforgettable characters – it's fun for the whole family!. <ITEM_END>

The model learns to associate key elements from the description with the item representation, enhancing its understanding of the item's characteristics beyond just the title. For instance, it can link concepts like "Walt Disney", "action-packed", "hilarious", and "whole family" to the movie title, providing a richer context for recommendations.

Unlike previous approaches [17] that use item reviews for item representation learning, our method utilizes item descriptions. This choice is motivated by the fact that product descriptions often provide more concise and relevant information about the item's features and intended use [29] whereas product reviews can be subjective and may contain irrelevant personal anecdotes [52]. We employ the Multiple Negative Ranking Symmetric Loss function [14] for this pretraining phase, as detailed in Section 4.4.3. This loss function serves two purposes: it brings the item representation closer to its description in the embedding space while simultaneously aligning the description representation with the item. This bidirectional optimization contributes to the model's ability to generalize and create universal representations, understanding what aspects of natural text to focus on when generating item representations.

4.4.1.2 Sequence-Sequence Contrastive Pretraining

To address the limited sequence awareness identified as a key weakness of the ST5 model, we implement a Sequence-Sequence contrastive learning phase inspired by the CL4SRec model's pretraining objective [55]. This pretraining step aims to provide our model with a robust sense of sequence, which is essential for effective sequential recommendation. Our approach enhances the model's ability to handle variations and inconsistencies in user interaction data while maintaining adaptability in sequence interpretation. To achieve this, we employ a series of carefully crafted sequence augmentations: item crop (s_{crop}), item reorder (s_{reorder}), and item drop (s_{drop}).

Let $s = (i_1, i_2, \ldots, i_n)$ represent an original user sequence, where i_k denotes the k-th item in the sequence, and n is the length of the sequence. To encourage the model to capture overall patterns rather than relying strictly on exact order, accommodating scenarios where users might interact with variants of the same item in flexible order, we implement the item reorder augmentation. Item reorder is expressed as $s_{\text{reorder}} = (i_1, \ldots, i_j, \pi(i_{j+1}, \ldots, i_{j+m}), i_{j+m+1}, \ldots, i_n)$, where π represents a random permutation and $m \leq \beta \times n$, introducing local shuffling within a sequence. β determines the maximum proportion of the sequence to reorder, and m is the length of the reordered subsequence.

The item crop augmentation simulates incomplete user histories by cropping a continuous portion of the sequence. It can be defined as $s_{\text{crop}} = (i_j, i_{j+1}, \ldots, i_{j+m})$,

where $1 \leq j \leq n$ and $m \leq \eta \times n$. η controls the maximum proportion of the sequence to crop, and m is the length of the cropped subsequence. This helps the model learn from partial sequences, improving its ability to make recommendations even with limited historical data.

To enhance the model's robustness to incomplete data and ensure that it can infer user preferences from partial information, we implement the item drop augmentation. In this technique, we randomly remove a subset of items $(i_{k_1}, i_{k_2}, \ldots, i_{k_m})$ from the sequence s, resulting in $s_{\text{drop}} = s \setminus (i_{k_1}, i_{k_2}, \ldots, i_{k_m})$, where $m \leq \delta \times n$ and m is the number of items dropped from the sequence. δ controls the maximum proportion of items to drop.

Formally, let $f(\cdot)$ denote our encoder function. We aim to minimize the distance between $f(s)$ and $f(s_{\text{aug}})$, where s_{aug} is randomly chosen from $\{s_{\text{crop}}, s_{\text{reorder}}, s_{\text{drop}}\}$. In our setup, (s, s_{aug}) forms the positive pair, while negative pairs are randomly sampled from other sequences within the batch using the Multiple Negative Symmetric Loss function [14], as detailed in Section 4.4.3.

A key distinction of our approach from the CL4SRec [55] lies in our contrastive learning setup. While CL4SRec contrasts between two augmented views of a sequence, we use the original sequence as an anchor and contrast it with its augmented version. This strategy serves a dual purpose: it teaches the model to maintain a consistent understanding of user preferences despite perturbations, while also preserving the essence of the original sequence. By contrasting with the original sequence, we ensure that the model learns transformations that are semantically meaningful and relevant to the task of sequential recommendation.

Through this carefully designed Sequence-Sequence contrastive learning phase, we equip our model with a nuanced understanding of user sequences. The model becomes adept at handling noisy, incomplete, or slightly reordered interaction data, making it more robust and flexible in real-world recommendation scenarios where user behavior can be inherently variable and unpredictable.

4.4.2 Fine tuning (Sequence-Item Contrastive Learning)

The final phase of our model development directly addresses the core objective of our reformulated sequential recommendation task: optimizing the alignment between user sequences and candidate items within a sentence retrieval framework. This final training stage builds on what the model learned earlier, combining its understanding of how users behave with its knowledge of item features.

The primary goal of fine-tuning is to teach the model how users interact with items, creating a unified understanding that bridges user behavior and item attributes. This is achieved through contrastive learning, where we pair user sequences s with their corresponding ground truth items i_{gt} to form positive pairs (s, i_{gt}), with negative pairs randomly sampled from the batch. Continuing our approach from the pretraining phases, we employ the Multiple Negative Ranking Symmetric Loss function [14], which is discussed in 4.4.3.

This loss function is particularly well-suited for our task as it optimizes bidirectional relationships, encouraging both the retrieval of relevant items given a user sequence and the identification of similar users given an item query. This symmetric approach ensures that the model learns to generate versatile representations effective for both user-to-item and item-to-user retrieval tasks. Optimizing the embedding space with this objective, aligns with our goal of developing universal representations, potentially enhancing the model's applicability to diverse recommendation tasks and scenarios beyond just sequential recommendation.

4.4.3 Loss Function

For all our training procedures, we utilize the Multiple Negative Ranking Symmetric Loss function [14]. This loss function is designed to bring similar sentence embeddings closer together while pushing dissimilar ones apart, thereby creating an effective representational space for our recommendation task. It operates on a batch of N sentence pairs $(S_{a_i}, S_{p_i})_{i=1}^{N}$, where S_a represents the anchor sentences and S_p represents the positive sentences we want to bring closer to the anchor. The function leverages in-batch negatives for efficient and effective contrastive learning. Let $f(\cdot)$ be our ST5 model function that generates embeddings for input sentences. For a given sentence pair (S_a, S_p), we have: $f(S_a) = \text{emb}_a \quad \text{and} \quad f(S_p) = \text{emb}_p$. The similarity between two embeddings is measured using cosine similarity. We compute the similarity scores between each anchor and every positive sentence, storing these scores in a matrix $\mathbf{S} \in R^{N \times N}$, where each entry S_{ij} represents the similarity between the i-th anchor sentence S_{a_i} and the j-th positive sentence S_{p_j}, scaled by a factor τ:

$$S_{ij} = \tau \cdot \text{sim}(\text{emb}_{a_i}, \text{emb}_{p_j}) \tag{4.1}$$

The loss function for a given batch is defined in two parts: forward loss and backward loss. The forward loss L_{forward} focuses on the similarity between the anchor embedding

emb_a and the positive embeddings emb_p in the batch, computed using the cross-entropy loss function:

$$L_{\text{forward}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp(S_{ii})}{\sum_{j=1}^{N}\exp(S_{ij})}\right) \tag{4.2}$$

The backward loss L_{backward} reverses the roles of anchor and positive sentences, focusing on the similarity between each positive embedding emb_p and the anchor emb_a, computed as:

$$L_{\text{backward}} = -\frac{1}{N}\sum_{j=1}^{N}\log\left(\frac{\exp(S_{jj})}{\sum_{i=1}^{N}\exp(S_{ji})}\right) \tag{4.3}$$

where S_{ji} represents the similarity score between positive S_{p_i} and anchor S_{a_j}.

The total symmetric loss combines the forward and backward losses by taking their average, ensuring that both anchor-to-positive and positive-to-anchor directions are optimized. This leads to robust and versatile embeddings that can capture complex relationships from both item-to-user and user-to-item perspectives, creating a universal representation space for recommendation tasks.

4.5 ST5-Final Model Performance Analysis

Looking at Figure 4.3 (Table in Appendix A.1), we observe several compelling insights that demonstrate the significant improvements achieved by our ST5-Final model over its predecessors and competitors. The most striking observation is the consistent superiority of ST5-Final across almost all categories and metrics. Our new model not only surpasses its zero-shot counterpart, ST5-Only, but also outperforms the previously leading UniSRec(BLaIR) model by substantial margins, despite UniSRec(BLaIR)'s strong performance in initial experiments.

The magnitude of these improvements is remarkable. In some instances, such as the *Baby Products* category, we see an astounding 121% increase in NDCG@10 (Table in Appendix A.1) over the second-best model. Moreover, an interesting pattern emerges when comparing the improvements in HR and NDCG metrics. Consistently, we observe larger percentage increases in NDCG compared to HR. This suggests that beyond just recommending relevant items, our new model has become significantly better at ranking these items in a way that aligns with user preferences, which is crucial for enhancing the user experience.

The strong performance of ST5-Final across diverse categories provides compelling evidence for the effectiveness of our text-based retrieval approach to recommendation

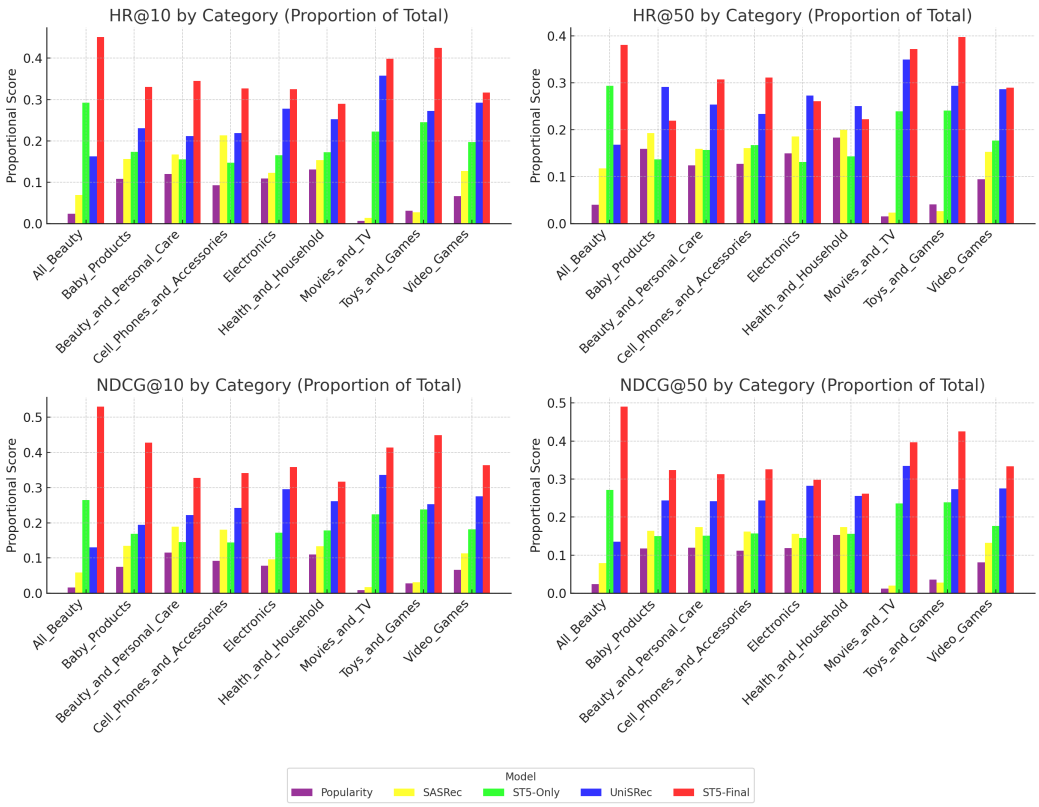

Figure 4.3: Comparison of Model Performance by Category. This figure displays four bar charts, each representing a metric.The bars are normalized to show the proportion of each model's contribution within each category

tasks. For instance, the ST5-Final model showed a significant improvement in *Electronics* and *Cell Phones and Accessories* categories where ST5-Only previously struggled. This indicates that our training process has successfully addressed the limitations we identified in the ST5-Only model.

By converting the recommendation problem into a sentence retrieval task, we created a more flexible and powerful model that can adapt to various product domains more effectively than traditional ID-based methods. In the following chapter, we will conduct experiments not only to analyze the effect of our pretraining in addressing the limitations of ST5-Only but also to prove the universality of our model-generated user sequence and item representation for cross-domain, cross-platform, and tasks beyond sequential recommendation, such as rating prediction.

Chapter 5

Experiment and Results

To thoroughly evaluate the effectiveness of our proposed ST5 model, we designed a comprehensive set of experiments addressing specific research questions. Each experiment aims to assess different aspects of the model's performance, from the impact of pretraining to its cross-domain capabilities and potential for diverse recommendation tasks. In this chapter, we describe these experiments and analyze their results.

5.1 Experiments Overview

5.1.1 Impact of Pretraining on Model Performance

Our first experiment aims to understand how different pretraining strategies affect the model's performance in sequential recommendation tasks. We compared three variants of our model across nine diverse datasets representing different product categories. The first variant, **ST5-FineTune**, undergoes only fine-tuning (Sequence-Item contrastive) without any pretraining and serves as our baseline to understand the impact of pretraining. The second variant, **ST5-ItemPre**, incorporates Item-Description contrastive pretraining and fine-tuning. The third variant, **ST5-Final**, is our proposed model, which undergoes both Item-Description and Sequence-Sequence contrastive pretraining and fine-tuning. This variant allows us to assess the cumulative effect of both pretraining phases.

By comparing these three variants, we aim to quantify the contribution of each pretraining phase to the model's overall performance. We hypothesize that ST5-Final will consistently outperform the other variants, validating our assumption that both pretraining phases are crucial for generating robust and versatile representations for

sequential recommendation tasks.

5.1.2 Quality of Learned Representations

We conducted a t-SNE (t-Distributed Stochastic Neighbor Embedding) [48] analysis on both item and user representations. This visualization technique allows us to project high-dimensional embeddings into a 2D space while preserving local relationships. We sampled items and users from all nine domains included in the dataset, which allowed us to observe both within-domain and cross-domain relationships.

We examined the UniSRec (BLaIR), ST5-Only, and ST5-Final models for item representation. We compared the item representation from UniSRec (BLaIR) to assess how our pretraining method, which uses item descriptions, performs against their pretraining method based on item reviews. The ST5-Only model provided a baseline by showing the initial representational space prior to training, while the ST5-Final model demonstrated the changes in this space after the complete training process.

User representations were analyzed by comparing visualizations from the ST5-Only model and ST5-Final model. This comparison illustrates how the training process affects the model's understanding of user behaviors and preferences.

This visual analysis complements our quantitative experiments by providing intuitive insights into how our model understands the relationships between items and users. Meaningful clustering in the ST5-Final visualizations would suggest that our model not only successfully learned to capture important semantic information, but also that the representations of items and users can be effectively used for tasks such as user segmentation and item categorization.

5.1.3 Cross-Domain Generalization Capabilities

To evaluate the cross-domain capabilities of our model, we test its performance on three new categories that were not included in the training data: *Books*, *Digital Music*, and *Amazon Fashion*. These categories were chosen for their distinct characteristics, presenting unique challenges to our model's generalization abilities.

The *Books* category represents a domain with rich textual information and diverse subgenres, requiring the model to understand complex semantic relationships within textual descriptions. *Digital Music*, while also content-driven, presents different user consumption and user engagement patterns, testing the model's ability to adapt to varied interaction behaviors. *Amazon Fashion* introduces a heavily visual and trend-dependent

domain, with unique seasonal patterns and style preferences, challenging the model's capacity to infer relevant features from item descriptions that may emphasize visual attributes.

By testing our model across these varied domains, we aim to rigorously assess its ability to transfer knowledge and make meaningful recommendations in diverse product spaces. This experiment will reveal whether our pretraining strategies have indeed created a model with robust cross-domain capabilities.

5.1.4 Cross-Platform Generalization Capabilities

To assess our model's versatility and universal representation capabilities beyond e-commerce, we evaluated its performance on the Yelp (2018) dataset [60]. This dataset, containing over 5 million reviews, business data, ratings, and check-in information, presents unique challenges with its diverse content types and complex user interaction patterns.

Unlike e-commerce datasets where user interactions primarily involve purchases or product views, the Yelp dataset focuses on service-based recommendations. We leveraged business descriptions and addresses as item features, while user review histories served as interaction sequences. This cross-domain evaluation is crucial for determining whether our model, initially trained on e-commerce data, can generate effective recommendations in a significantly different context. Strong performance on the Yelp dataset would indicate that our pretraining strategies have successfully captured fundamental aspects of user preferences and item characteristics, needed for the recommendation task.

5.1.5 Effectiveness in Rating Prediction Task

To check the universality of the representations given by our model, we applied them to a rating prediction task. This experiment aims to demonstrate that our learned representations are effective in various downstream recommendation tasks, beyond just next-item prediction.

We use the user's interaction history to predict their rating for a new item on a scale of 1 to 5. Two variants of our model, ST5-Only and ST5-Final, are compared against established baselines: Mean Rating and Probabilistic Matrix Factorization (PMF) [28]. The Mean Rating baseline simply predicts the average rating for each item across all users, providing a lower bound for performance. PMF factorizes the

user-item interaction matrix into lower-dimensional user and item latent factors, known for handling sparse data and providing personalized predictions.

For our ST5 models, we generate user and item sentence embeddings from interaction history text and item descriptions, respectively. These embeddings are then concatenated and fed into a dense neural network, which performs regression to predict ratings. Performance is evaluated using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Strong performance in this task, particularly from the ST5-Final model, would indicate that our pretraining strategies have indeed created universal representations that capture user preferences and item characteristics, making them valuable for a variety of recommendation tasks.

5.1.6 Performance Variation with User History Length

To better understand how ST5-Only and ST5-Final models handle different lengths of user interaction histories, we conducted an in-depth analysis of our models' performance as a function of interaction history length. This experiment assesses how effectively our pretraining strategies, particularly the Sequence-Sequence contrastive pretraining, address the sequence modeling limitations inherent in the ST5-Only architecture.

In this experiment, we examined how our models perform with different numbers of items in a user's interaction sequence. We created subsets of users by filtering them based on the length of their interaction history, allowing us to analyze performance across various history lengths. To capture the effect of different user behaviors across various domains, we selected three distinct product categories: *All Beauty*, *Video Games*, and *Baby Products*. These categories were chosen to represent diverse user interaction patterns and preferences, allowing us to assess the effect of different user behaviors in different domains. We also conduct a comparative analysis, contrasting our model's performance against baselines for different history lengths.

5.2 Result Analysis

5.2.1 Impact of Pretraining on Model Performance

Our comprehensive analysis of the performance improvements across model iterations ST5-Only, ST5-NoPre, ST5-ItemPre, and ST5-Final, reveals compelling evidence of the effectiveness of our pretraining strategies in enhancing sequential recommendation tasks. As illustrated in Figure 5.1, the results demonstrate a clear progression

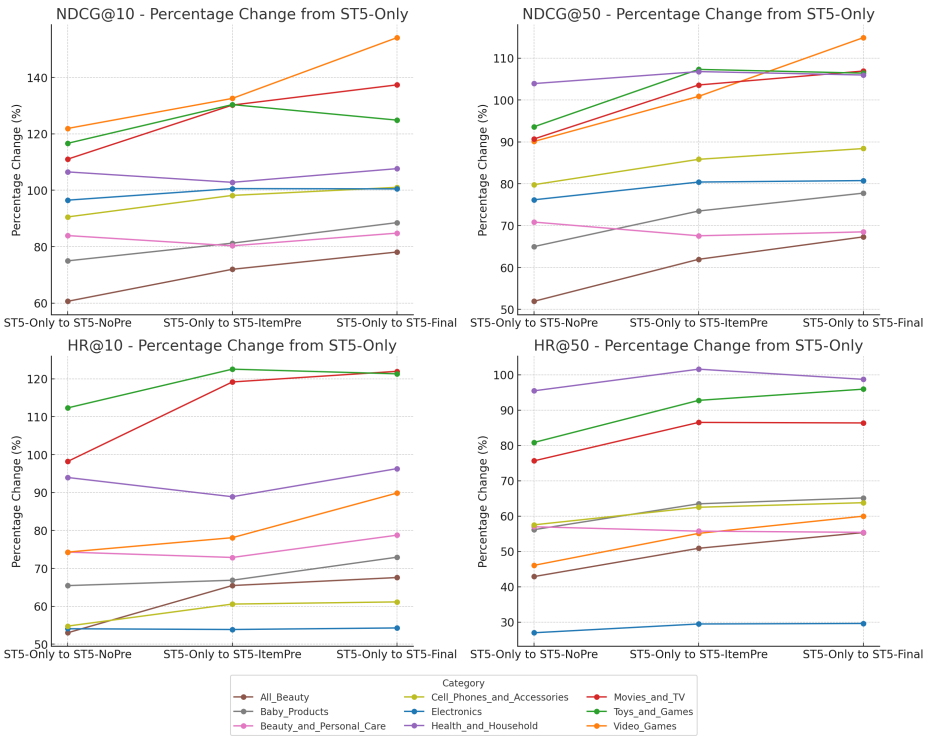

Figure 5.1: Performance improvements from ST5-Only to ST5-NoPre, ST5-ItemPre, and ST5-Final models for NDCG and HR metrics across product categories.

in model performance, with each iteration addressing specific limitations identified earlier. The most significant improvement was observed in the transition from ST5-Only to ST5-NoPre, yielding a remarkable 77.87% mean improvement across all metrics and categories (Table in Appendix A.2). This substantial gain directly addresses the weakness of the ST5 model discussed in Section 4.3, thus highlighting the importance of domain-adapting the ST5 model for sequential recommendation tasks.

Subsequent pretraining phases further refined the model's capabilities. The transition to ST5-ItemPre resulted in a 3.70% mean improvement, validating our Item-Description contrastive pretraining approach. The final iteration to ST5-Final, incorporating Sequence-Sequence contrastive pretraining, achieved an additional 1.88% mean improvement, effectively addressing the limitation of ST5's limited sequence awareness. Additionally, improvements in NDCG scores were more prominent than HR scores, suggesting that our pretraining phases enhanced not only overall performance but also the quality of recommendation rankings.

These results demonstrates the success of our two-phase pretraining approach in transforming a general-purpose language model into a powerful, task-specific recommendation engine. By systematically addressing key limitations, we have significantly

advanced the capabilities of our model in sequential recommendation tasks.

5.2.2 Quality of Learned Representations

Our analysis of item and user representations across three models—ST5-Only, UniSRec (BLaIR), and ST5-Final—reveals significant improvements in semantic understanding and domain separation achieved by our proposed approach, as illustrated in Figures 5.2 and 5.3.

5.2.2.1 Item Representations

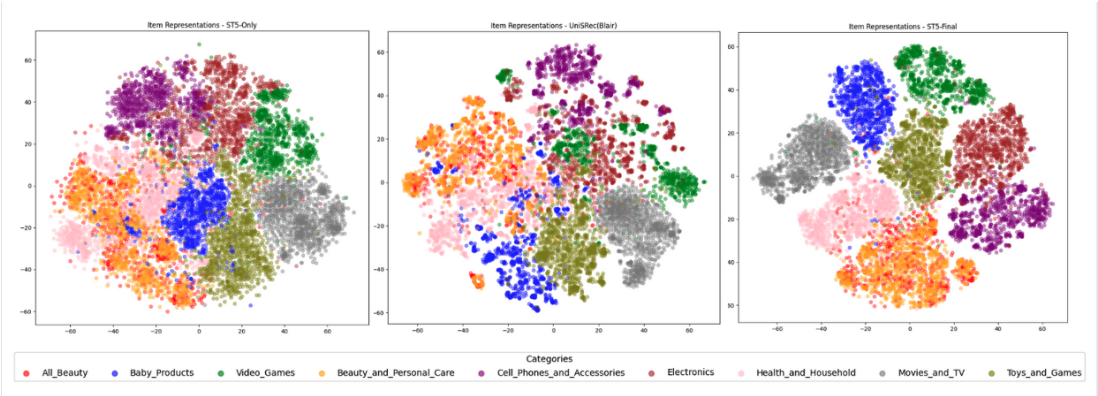

Figure 5.2: Item Representation for ST5-Only, UniSRec (BLaIR), and ST5-Final in order

The ST5-Only model, while showing some semantic understanding, exhibits considerable overlap between different product categories, supporting our earlier hypothesis regarding the semantic gap between general language understanding and specific recommendation tasks. In contrast, ST5-Final demonstrates remarkably clear clustering and boundaries between different product domains, showcasing its ability to capture nuanced product characteristics. For instance, the model effectively distinguishes between closely related categories like *Electronics* and *Cell Phones and Accessories* while recognizing their technical similarities.

Compared to the UniSRec (BLaIR) baseline, our ST5-Final model shows superior performance in creating distinct and meaningful item representations. While UniSRec (BLaIR) exhibits good clustering, it lacks the clear separation and nuanced relationships evident in our approach. The ST5-Final model's visualization reveals more compact clusters with sharper boundaries between dissimilar categories, while still maintaining logical proximity between related domains. This improved separation and grouping suggest that our Item-Description contrastive pretraining method has indeed been more

effective in bridging the semantic gap, supporting our hypothesis about the advantages of using item descriptions in recommendation tasks.

5.2.2.2 User Representations

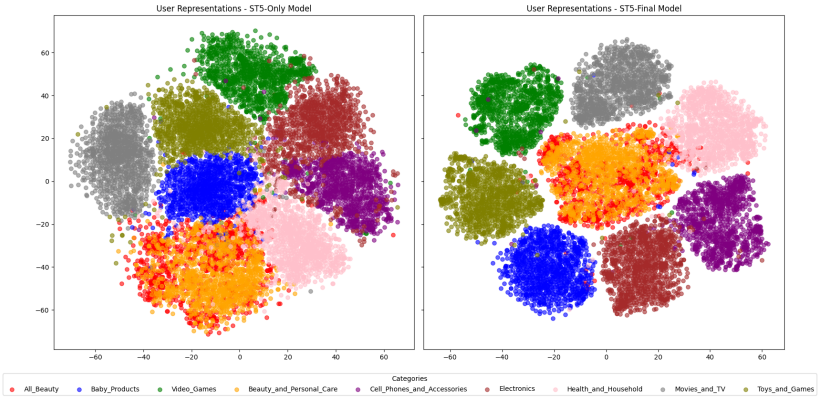

Figure 5.3: User Representation for ST5-Only and ST5-Final model in order.

The analysis of user representations reveals a stark contrast between the ST5-Only and ST5-Final models. While ST5-Only shows limited ability to differentiate between diverse user behaviors, ST5-Final exhibits well-defined boundaries between different domain clusters, with minimal outliers and more compact groupings. This improvement provides strong evidence that our pretraining strategy has significantly enhanced the model's capacity to distinguish and categorize diverse user behaviors.

ST5-Final captures semantic relationships in user behavior with remarkable intuition and nuance. For example, users interested in beauty-related categories form closely related clusters, while entertainment-related domains such as *Video Games* and *Toys and Games* form a loosely connected cluster. These intuitive groupings suggest that the model has developed a sophisticated understanding of complex user behaviors across various domains.

5.2.2.3 Comprehensive Analysis of the Representations

When we compare item and user representations across models, we notice several key improvements in the ST5-Final model. The ST5-Final model demonstrates consistent semantic relationships in both item and user representations, indicating the model captures semantics of how product categories and user behaviors interconnect. Our model also creates denser, more distinct clusters for both items and users, achieving clearer separation between unrelated categories while maintaining smoother transitions

between related ones. Additionally, we observe a substantial reduction in outliers for both item and user representations, suggesting more robust and reliable representations. These improvements collectively demonstrate that our pretraining strategies have effectively addressed the key weaknesses identified in the ST5-Only model and that our model can be used for tasks such as item and user segmentation.

The consistency and quality of representations across diverse product categories suggest that our model has developed a more universal understanding of both items and users. This universality holds significant promise for cross-domain and cross-platform recommendation capabilities, as well as for other recommendation tasks such as rating predictions. These capabilities are demonstrated in the following sections.

5.2.3 Cross-Domain Generalization Capabilities

Domain	Metrics	Pop	SASRec	UniSRec (BLaIR)	ST5-Only	ST5-Final
Amazon_Fashion	NDCG@10	0.35	0.42	1.07	1.64	**2.32**
	NDCG@50	0.55	0.69	1.32	1.92	**2.59**
	HR@10	0.57	0.87	1.94	2.69	**3.34**
	HR@50	1.54	2.23	3.09	3.92	**4.47**
Digital_Music	NDCG@10	0.14	0.61	0.84	4.04	**4.90**
	NDCG@50	0.22	0.91	1.54	4.95	**5.73**
	HR@10	0.32	0.84	1.81	5.66	**6.49**
	HR@50	0.65	2.27	5.12	9.25	**9.76**
Books	NDCG@10	0.04	0.56	**1.44**	1.12	1.38
	NDCG@50	0.08	0.60	**1.94**	1.51	1.82
	HR@10	0.11	0.79	**2.87**	2.25	2.66
	HR@50	0.26	1.01	**5.17**	3.96	4.60

Table 5.1: Cross-Domain Performance Comparison using various models. NDCG and HR scores are in percentage.

The results from our cross-domain experiment on *Books*, *Digital Music*, and *Amazon Fashion* categories provide compelling evidence of ST5-Final's robust generalization capabilities. As shown in Table 5.1, ST5-Final consistently outperforms ST5-Only across all three domains, demonstrating the effectiveness of our pretraining strategies in enhancing the model's ability to transfer knowledge to previously unseen product categories.

ST5-Final's performance is particularly impressive in the *Digital Music* category, where it significantly outperforms all other models. This suggests that our model has

successfully captured the nuances of content-driven recommendations and adapted well to the unique consumption patterns of this domain. Similarly, in the visually-oriented and trend-dependent *Amazon Fashion* category, ST5-Final shows strong performance, indicating its ability to infer relevant features from textual item descriptions even for products where visual attributes are crucial. The *Books* category presents a more challenging scenario for our model. While ST5-Final still shows improvement over ST5-Only, its performance is slightly behind UniSRec in this domain. This could be attributed to the large item corpus typical of book datasets, which may strain the model's capacity to distinguish between numerous similar items. Additionally, the limited input of book titles alone may not provide sufficient semantic context to capture the complex patterns of user preferences in the literature domain.

ST5-Final consistently improved both NDCG and HR metrics across all three new domains, demonstrating its ability to rank items more accurately and retrieve a broader range of relevant items. This performance, achieved without domain-specific training, showcases strong cross-domain capabilities. The text-based approach of ST5-Final appears to be particularly advantageous for cross-domain generalization, as textual representations can capture abstract concepts and features that are applicable across various product categories. This adaptability indicates strong potential for practical applications where the model must handle unfamiliar product categories and rapidly adjust to varied domains.

5.2.4 Cross-Platform Generalization Capabilities

Domain	Metrics	Pop	SASRec	UniSRec (BLaIR)	ST5-Only	ST5-Final
Yelp	NDCG@10	0.22	1.01	0.99	0.62	**1.37**
	NDCG@50	0.46	1.65	**1.78**	1.03	1.67
	HR@10	0.46	1.95	**1.96**	1.01	1.57
	HR@50	1.57	4.90	**5.65**	2.74	2.82

Table 5.2: Performance comparison for the Yelp Dataset using various models. NDCG and HR scores are in percentage.

The results from the Cross-Platform experiment demonstrate ST5-Final's strong performance on non-e-commerce recommendation tasks as well. As observed in Table 5.2, ST5-Final achieved the highest NDCG@10 score among all models tested, including domain-specific models like SASRec and UniSRec (BLaIR).

The substantial improvement from ST5-Only to ST5-Final across all metrics indicates the effectiveness of our pretraining strategies in bridging the semantic gap between general language understanding and specific recommendation tasks. By utilizing only textual data, ST5-Final exhibited robust cross-platform capabilities, performing competitively against models specifically trained for this task.

While UniSRec (BLaIR) shows better performance in metrics such as NDCG@50, HR@10, and HR@50, it is important to note that ST5-Final was not trained on Yelp dataset, yet its performance closely approaches that of UniSRec (BLaIR), which is particularly impressive. This suggests that our model has successfully captured fundamental aspects of user preferences and item characteristics that generalize beyond e-commerce, demonstrating its potential for use across various real-world platform recommendations.

5.2.5 Effectiveness in Rating Prediction Task

Model	RMSE	MAE
Mean	1.460	1.113
PMF	1.394	<u>1.072</u>
ST5-Only	<u>1.387</u>	1.091
ST5-Final	**1.385**	**1.061**

Table 5.3: Comparison of RMSE and MAE across different models.

The results from our rating prediction experiment provide compelling evidence for the universality and effectiveness of our ST5 model's representations. Both ST5-Only and ST5-Final outperform traditional baselines (Mean and PMF) in terms of RMSE, demonstrating that our text-based approach captures more nuanced user preferences and item characteristics than conventional methods, even in a task it wasn't explicitly trained for.

The strong performance of ST5-Only in a zero-shot setting is particularly noteworthy. Its ability to outperform PMF in RMSE without any task-specific fine-tuning demonstrates the effectiveness of converting recommendation problems into textual tasks, suggesting potential applicability across a wide range of recommendation scenarios.

ST5-Final's superior performance across both RMSE and MAE metrics further validates our pretraining strategies. Perhaps most significantly, the model's strong

performance in this explicit feedback task (rating prediction), despite being primarily trained on implicit feedback (next-item prediction), underscores the universality of the learned representations. This generalization across feedback types strongly supports our hypothesis that the representations learned by our ST5 model are indeed universal and valuable for recommendation scenarios beyond just sequential recommendation.

5.2.6 Performance Variation with User History Length

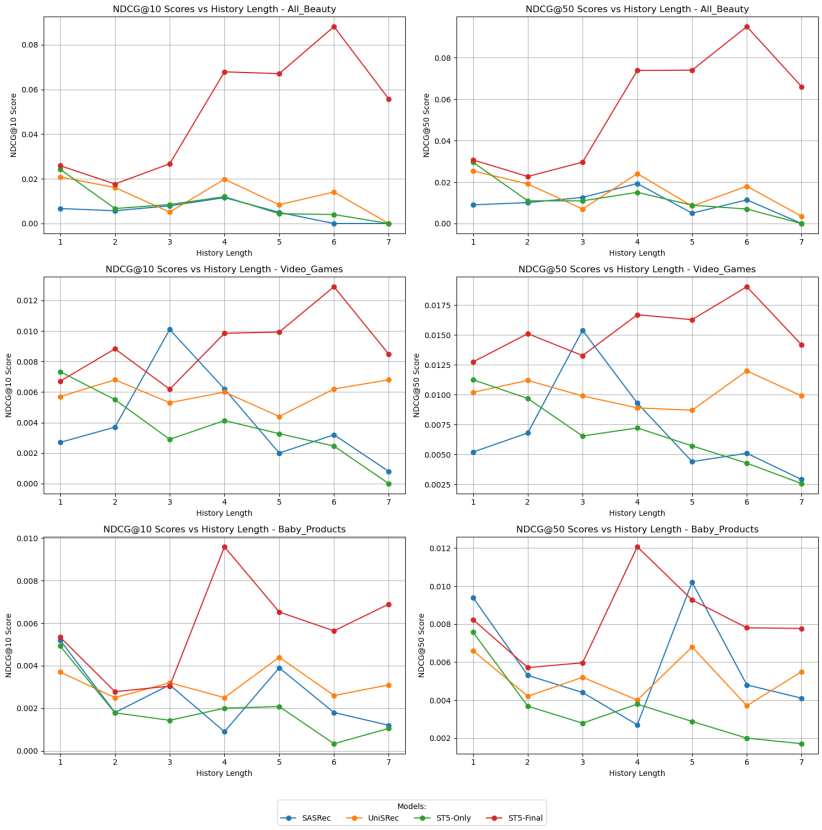

Figure 5.4: Performance Variation of Various Models with User Interaction History Length.

The findings of our experiment evaluating the performance of various recommendation models across different user interaction history lengths reveal intriguing patterns that highlight the effectiveness of our proposed ST5-Final model, as illustrated in figure 5.4.

Our analysis demonstrates that the ST5-Final model consistently outperforms all other models across various product categories and history lengths. This superior performance is particularly evident as the number of interacted items in user history increases. For instance, in the *Baby Products* category, ST5-Final shows a marked

improvement in NDCG scores as history length grows, with a notable peak at a history length of 4. Similar trends are observed in the *All Beauty* and *Video Games* categories. A particularly noteworthy aspect of ST5-Final's performance is its effectiveness in cold-start scenarios for users with 1-2 interacted items, where ST5-Final consistently achieves the highest performance. This indicates that our task-based formulation and model architecture have successfully addressed one of the most challenging aspects of recommendation systems.

A compelling finding emerges when comparing ST5-Final to the ST5-Only model. While both models exhibit similar performance for single-interaction histories, their trajectories diverge significantly as the number of interacted items increases. ST5-Final demonstrates a rapid performance improvement, whereas ST5-Only's performance tends to decline. This divergence suggests that ST5-Only primarily relies on similarity-based recommendations derived from single interactions, essentially reducing the task to a sentence similarity problem. In contrast, ST5-Final, enhanced with Sequence-Sequence contrastive learning, overcomes this limitation. This validates our hypothesis that our pretraining strategies effectively capture temporal patterns in user behavior.

Interestingly, we observed varying performance trends across different domains, which aligns with findings in previous research [35]. For instance, in the *Baby Products* category, we noticed a general trend of all models decreasing in performance up to 3-4 interacted items, then increasing at 4-5 items, before decreasing again. In *All Beauty* and *Video Games*, SASRec's performance increases up to a certain point and then decreases, possibly due to longer histories introducing confusion. Despite these variations, SASRec tends to perform better than ST5-Only for longer histories, highlighting the importance of sequence modeling in recommendation tasks. In contrast, UniSRec (BLaIR) maintains relatively consistent performance across increasing numbers of items in history, likely due to its ability to generate universal user representations. This stability sets UniSRec (BLaIR) apart from the more variable performance of SASRec and ST5-Only.

The ST5-Final model's ability to maintain and even improve performance with increasing history length demonstrates its robustness and versatility across various recommendation scenarios, from cold-start to long-term user interactions. These results collectively underscore the success of our approach in leveraging extended user interactions and capturing complex temporal patterns in user behavior.

Chapter 6

Conclusion

In this chapter, we summarise our project's limitations, propose future research directions, outline key implications, and provide an overall summary of our findings.

6.1 Limitations and Future Work

Although our research has shown promising results, several limitations present opportunities for future work. A significant constraint is the ST5's current input limit of 255 tokens, restricting its ability to process longer user histories that could improve recommendations [33]. Future work could explore expanding the model's context size by implementing chunking strategies [21] or adopting approaches such as long text encoding with attention sparsity, as presented by Liu et al. [27]

Our current implementation utilizes the ST5's base variant, but the authors of ST5 have demonstrated a scaling effect with increased parameters. Fine-tuning larger models could potentially enhance performance significantly. However, this approach raises concerns about inference time. Moreover, Qu et. al [34] suggest that many encoder layers may be redundant for sequential recommendation tasks when using an LLM. To address these issues, future work could investigate training the model with 2D Matryoshka embedding loss [26], which has shown potential to train an encoder model in a way that can result in comparable performance even with fewer encoder layers and smaller embedding sizes. Alternatively, exploring the effects of embedding quantization [57] could help reduce the model's memory footprint without sacrificing performance. Additionally, future research could explore parameter-efficient techniques like LoRA [19] to reduce training time and lower computational resources required.

While our model has shown promising results for non-e-commerce datasets, there's

still room for improvement in this area. In the future, we could train on a more diverse range of platforms beyond e-commerce, including datasets from platforms like Steam [22], Netflix [31], Twitch [38], and MyAnimeList [47], to make our recommendations more universally applicable. Additionally, incorporating a broader spectrum of user interactions beyond transactional data, could help our model develop a more comprehensive understanding of user behavior [13]. Another potential avenue for future work is to incorporate multidomain data [11], such as images, especially for improving recommendations in visually driven domains like Clothing, Shoes, and Jewelry.

Lastly, a crucial area for future development lies in explainable recommendations [62]. Enhancing our model to provide clear rationales for its suggestions would not only improve user trust and satisfaction but also offer valuable insights into the recommendation process. This direction aligns with the growing demand for transparency and ethical AI systems.

6.2 Final Remarks

In conclusion, our research presents a promising direction in sequential recommendation by reformulating it as a sentence retrieval task using the Sentence-T5 model. This approach has demonstrated significant improvements over strong baseline models like SASRec and UniSRec(BLaIR) across diverse product categories. The ST5-Final model, developed through our two-phase pretraining strategy, effectively bridges the semantic gap between natural language understanding and recommendation tasks while capturing complex sequential patterns in user behavior. The model's strong performance on unseen product categories and non-e-commerce datasets showcases its broad applicability across recommendation domains. This adaptability could reduce the need for frequent model retraining in new domains, potentially leading to more sustainable AI systems.

The text-based nature of our approach opens up new possibilities for creating more inclusive and interpretable recommendation systems, which can cater to a wider range of user needs and preferences, thus delivering more personalized recommendations. Moreover, our model's strong performance in cold-start scenarios can be particularly valuable during user onboarding processes, enabling more engaging and personalized experiences for new users. Furthermore, our model's ability to generate universal user and item representations that are effective even for other recommendation tasks like rating prediction suggests our work could contribute towards developing a foundation model for recommendation, a long-standing goal in the field.

Bibliography

[1] Amazon. Amazon, 2024. Accessed: 2024-08-20.

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[3] Tesfaye Fenta Boka, Zhendong Niu, and Rama Bastola Neupane. A survey of sequential recommendation systems: Techniques, evaluation, and future directions. *Information Systems*, page 102427, 2024.

[4] Tom B Brown. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*, 2020.

[5] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084*, 2022.

[6] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[7] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, 39(1):1–42, 2020.

[9] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*, 2023.

[10] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315, 2022.

[11] Shijie Geng, Juntao Tan, Shuchang Liu, Zuohui Fu, and Yongfeng Zhang. Vip5: Towards multimodal foundation models for recommendation. *arXiv preprint arXiv:2305.14302*, 2023.

[12] Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008*, 2018.

[13] Google Research. Transformers in music recommendation, August 10 2023. Accessed: 2024-08-20.

[14] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply, 2017.

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

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

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

[18] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pages 364–381. Springer, 2024.

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[20] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[21] Mandar Joshi, Omer Levy, Daniel S Weld, and Luke Zettlemoyer. Bert for coreference resolution: Baselines and analysis. *arXiv preprint arXiv:1908.09091*, 2019.

[22] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[23] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1258–1267, 2023.

[26] Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2d matryoshka sentence embeddings, 2024.

[27] Zhenghao Liu, Sen Mei, Chenyan Xiong, Xiaohua Li, Shi Yu, Zhiyuan Liu, Yu Gu, and Ge Yu. Text matching improves sequential recommendation by reducing popularity biases. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1534–1544, 2023.

[28] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

[29] Muhammad Ghiffary Mokobombang and Nurrani Kusumawati. The impact of product description, product photo, rating, and review on purchase intention in e-commerce. *Journal of Consumer Studies and Applied Marketing*, 1(2):137–147, 2023.

[30] Shanlei Mu, Yupeng Hou, Wayne Xin Zhao, Yaliang Li, and Bolin Ding. Id-agnostic user behavior pre-training for sequential recommendation. In *China Conference on Information Retrieval*, pages 16–27. Springer, 2022.

[31] Netflix, Inc. Netflix prize data, 2009. Accessed: 2024-08-20.

[32] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*, 2021.

[33] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2671–2679, 2019.

[34] Zekai Qu, Ruobing Xie, Chaojun Xiao, Xingwu Sun, and Zhanhui Kang. The elephant in the room: Rethinking the usage of pre-trained language model in sequential recommendation. *arXiv preprint arXiv:2404.08796*, 2024.

[35] Zekai Qu, Ruobing Xie, Chaojun Xiao, Yuan Yao, Zhiyuan Liu, Fengzong Lian, Zhanhui Kang, and Jie Zhou. Thoroughly modeling multi-domain pre-trained recommendation as language. *arXiv preprint arXiv:2310.13540*, 2023.

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[38] Jérémie Rappaz, Julian McAuley, and Karl Aberer. Recommendation on live-streaming platforms: Dynamic availability and repeat consumption. In *Fifteenth ACM Conference on Recommender Systems*, pages 390–399, 2021.

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

[40] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.

[41] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[42] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009(1):421425, 2009.

[43] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[44] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573, 2018.

[45] The University of Edinburgh. Edinburgh compute and data facility (ecdf), 2024. Accessed: 2024-05-01.

[46] Poonam B Thorat, Rajeshwari M Goudar, and Sunita Barve. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110(4):31–36, 2015.

[47] Cooper Union. Anime recommendations database, 2017. Accessed: 2024-08-20.

[48] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] Lei Wang and Ee-Peng Lim. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153*, 2023.

[51] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830*, 2019.

[52] Lu Wei, Shufan Ma, and Maoze Wang. Understanding the information characteristics of consumers' online reviews: the evidence from chinese online apparel shopping. *Electronic Commerce Research*, pages 1–27, 2023.

[53] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*, 2023.

[54] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 346–353, 2019.

[55] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE, 2022.

[56] 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. Recent advances in recbole: Extensions with more practical considerations, 2022.

[57] Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. Efficient passage retrieval with hashing for open-domain question answering. *arXiv preprint arXiv:2106.00882*, 2021.

[58] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion proceedings of the web conference 2020*, pages 441–447, 2020.

[59] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*, 2019.

[60] Yelp. Yelp open dataset. https://www.yelp.com/dataset, 2024. Accessed: 2024-08-19.

[61] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001*, 2023.

[62] Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

[63] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. Language models as recommender systems: Evaluations and limitations. In *NeurIPS 2021 Workshop on I (Still) Can't Believe It's Not Better*, 2021.

[64] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. Recbole 2.0: Towards a more up-to-date recommendation library. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4722–4726, 2022.

[65] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*, pages 4653–4664. ACM, 2021.

[66] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*, 2023.

[67] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902, 2020.

[68] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. Using temporal data for making recommendations. *arXiv preprint arXiv:1301.2320*, 2013.

Appendix A

Experiments Results Table

Category	Metric	Pop	SASRec	UniSRec(BLaIR)	ST5-Only	ST5-Final	Improvement%
All Beauty	H@10	0.38	1.08	2.55	4.59	**7.08**	54.27
	H@50	0.98	2.90	4.14	7.25	**9.40**	29.60
	N@10	0.18	0.64	1.42	2.89	**5.79**	100.52
	N@50	0.31	1.02	1.75	3.51	**6.35**	80.78
Baby Products	H@10	0.50	0.72	1.06	0.80	**1.52**	89.86
	H@50	2.09	2.53	**3.81**	1.80	2.87	60.02
	N@10	0.20	0.36	0.52	0.45	**1.15**	153.88
	N@50	0.53	0.74	1.10	0.68	**1.46**	114.90
Video Games	H@10	0.50	0.95	2.19	1.47	**2.37**	61.06
	H@50	1.73	2.80	5.24	3.23	**5.29**	63.81
	N@10	0.29	0.49	1.19	0.78	**1.58**	101.15
	N@50	0.55	0.89	1.85	1.19	**2.24**	88.37
Movies and TV	H@10	0.05	0.12	3.00	1.87	**3.34**	78.75
	H@50	0.23	0.35	5.18	3.55	**5.52**	55.38
	N@10	0.04	0.08	1.51	1.01	**1.86**	84.79
	N@50	0.07	0.12	1.99	1.40	**2.36**	68.52
Toys and Games	H@10	0.14	0.12	1.20	1.08	**1.87**	72.85
	H@50	0.44	0.28	3.08	2.53	**4.18**	65.14
	N@10	0.06	0.07	0.57	0.54	**1.01**	88.45
	N@50	0.13	0.10	0.98	0.86	**1.52**	77.62
Beauty and Personal Care	H@10	0.49	0.68	0.86	0.63	**1.40**	121.20
	H@50	1.14	1.46	2.33	1.44	**2.82**	95.96
	N@10	0.25	0.40	0.47	0.31	**0.70**	124.92
	N@50	0.39	0.56	0.78	0.49	**1.01**	106.14
Health and Household	H@10	0.48	0.56	0.92	0.63	**1.06**	67.46
	H@50	1.87	2.04	**2.55**	1.46	2.27	55.31
	N@10	0.20	0.24	0.47	0.32	**0.57**	78.09
	N@50	0.49	0.56	0.82	0.50	**0.84**	67.47
Cell Phones and Accessories	H@10	0.33	0.75	0.77	0.52	**1.15**	121.97
	H@50	1.14	1.45	2.10	1.50	**2.80**	86.36
	N@10	0.16	0.32	0.43	0.25	**0.60**	137.33
	N@50	0.33	0.48	0.72	0.46	**0.96**	106.91
Electronics	H@10	0.29	0.33	0.75	0.45	**0.88**	96.19
	H@50	1.19	1.47	2.16	1.04	**2.07**	98.66
	N@10	0.10	0.12	0.37	0.22	**0.45**	107.41
	N@50	0.28	0.37	0.67	0.34	**0.71**	106.11

Table A.1: Performance comparison using various models. The scores are expressed as percentages. ”H” stands for Hit Rate (HR), and ”N” stands for Normalized Discounted Cumulative Gain (NDCG). ”Improvement %” is the improvement of ST5-Final over its zero-shot counterpart ST5-Only Model. The best model is marked in bold and the second-best model is underlined.

Category	Metric	ST5-Only	ST5-NoPre	ST5-ItemPre	ST5-Final	Item2Desc Improv.%	Seq2Seq Improv.%
All Beauty	H@10	4.59	7.07	7.06	7.07	-0.13	0.26
	H@50	7.25	9.20	9.38	9.39	1.95	0.11
	N@10	2.89	5.68	5.80	5.79	2.08	-0.05
	N@50	3.51	6.19	6.33	6.35	2.41	0.19
Baby Products	H@10	0.80	1.39	1.42	1.52	2.18	6.61
	H@50	1.80	2.62	2.79	2.87	6.19	3.14
	N@10	0.45	1.00	1.05	1.15	4.79	9.26
	N@50	0.68	1.29	1.36	1.46	5.66	6.97
Video Games	H@10	1.47	2.28	2.37	2.37	3.75	0.35
	H@50	3.23	5.09	5.25	5.29	3.16	0.80
	N@10	0.78	1.49	1.55	1.58	4.00	1.44
	N@50	1.19	2.13	2.21	2.24	3.37	1.38
Movies and TV	H@10	1.87	3.26	3.23	3.34	-0.81	3.40
	H@50	3.55	5.57	5.53	5.52	-0.79	-0.22
	N@10	1.01	1.85	1.81	1.86	-1.97	2.49
	N@50	1.40	2.39	2.35	2.36	-1.91	0.56
Toys and Games	H@10	1.08	1.79	1.81	1.87	0.85	3.64
	H@50	2.53	3.95	4.14	4.18	4.68	1.03
	N@10	0.54	0.94	0.97	1.01	3.58	4.01
	N@50	0.86	1.41	1.49	1.52	5.15	2.46
Beauty and Personal Care	H@10	0.63	1.34	1.41	1.40	4.81	-0.55
	H@50	1.44	2.60	2.77	2.82	6.59	1.65
	N@10	0.31	0.67	0.71	0.69	6.33	-2.39
	ND@50	0.49	0.95	1.01	1.01	7.07	-0.43
Cell Phones and Accessories	H@10	0.52	1.03	1.13	1.15	10.58	1.28
	H@50	1.50	2.64	2.80	2.80	6.19	-0.08
	N@10	0.25	0.54	0.59	0.60	9.08	3.11
	N@50	0.46	0.89	0.95	0.96	6.76	1.63
Electronics	H@10	0.45	0.86	0.84	0.88	-2.62	3.92
	H@50	1.04	2.03	2.10	2.07	3.13	-1.43
	N@10	0.22	0.45	0.44	0.45	-1.79	2.39
	N@50	0.34	0.70	0.71	0.71	1.40	-0.40
Health and Household	H@10	0.63	0.96	1.04	1.06	8.14	1.27
	H@50	1.46	2.08	2.20	2.27	5.60	2.95
	N@10	0.32	0.52	0.55	0.57	7.06	3.56
	N@50	0.50	0.76	0.81	0.84	6.60	3.31
Mean Improvement	-	-	-	-	-	**3.70**	**1.88**

Table A.2: Performance Comparison of Different Variants of ST5. These are results for the experiment explained in Section 5.1.1. ”Item2Desc Improv. %” describes the improvement of ST5-ItemPre over ST5-NoPre. ”Seq2Seq Improv. %” describes the improvement of ST5-Final over ST5-ItemPre. The scores are expressed as percentages. ”H” stands for Hit Rate (HR), and ”N” stands for Normalized Discounted Cumulative Gain (NDCG).

Category	Metric	Struct	Unstruc	Inst.+Struct.	Inst+UnStruct.
All Beauty	N@10	2.85	2.71	1.82	1.74
	N@50	3.47	3.31	2.26	2.14
	H@10	4.57	4.46	3.16	3.03
	H@50	7.22	7.00	5.07	4.76
Baby Products	N@10	0.46	0.46	0.29	0.28
	N@50	0.70	0.67	0.48	0.44
	H@10	0.81	0.81	0.56	0.55
	H@50	1.89	1.76	1.39	1.25
Beauty and Personal Care	N@10	0.33	0.31	0.18	0.16
	N@50	0.52	0.48	0.31	0.30
	H@10	0.69	0.65	0.37	0.33
	H@50	1.55	1.41	0.94	0.93
Cell Phones and Accessories	N@10	0.27	0.23	0.12	0.11
	N@50	0.45	0.42	0.23	0.23
	H@10	0.55	0.50	0.29	0.24
	H@50	1.39	1.40	0.81	0.80
Electronics	N@10	0.22	0.21	0.14	0.13
	N@50	0.36	0.33	0.24	0.23
	H@10	0.46	0.45	0.30	0.27
	H@50	1.08	1.04	0.76	0.72
Health and Household	N@10	0.33	0.31	0.23	0.22
	N@50	0.50	0.47	0.36	0.36
	H@10	0.65	0.63	0.46	0.42
	H@50	1.42	1.36	1.08	1.03
Movies and TV	N@10	1.05	1.05	0.63	0.52
	N@50	1.45	1.47	0.93	0.78
	H@10	2.01	1.98	1.05	0.95
	H@50	3.77	3.78	2.37	2.06
Toys and Games	N@10	0.54	0.53	0.36	0.32
	N@50	0.85	0.84	0.63	0.58
	H@10	1.09	1.05	0.73	0.64
	H@50	2.50	2.46	1.96	1.79
Video Games	N@10	0.81	0.78	0.43	0.42
	N@50	1.24	1.21	0.78	0.74
	H@10	1.50	1.48	0.86	0.87
	H@50	3.39	3.34	2.38	2.25

Table A.3: Comparison of User Sequence Text Representation Methods. This table presents the results for different approaches to representing user sequences. ”Struct” refers to Structured, ”Unstruc” to Unstructured, ”Inst.+Struct.” to Instruction + Structured, and ”Inst+UnStruct.” to Instruction + Unstructured. ”N” stands for Normalized Discounted Cumulative Gain (NDCG), and ”H” stands for Hit Rate (HR).

Category	Item Text	N@10	N@50	H@10	H@50
All Beauty	Text	2.91	3.53	4.59	7.21
	Text+Desc.	2.12	2.64	3.58	5.88
Baby Products	Text	0.47	0.69	0.83	1.81
	Text+Desc.	0.25	0.38	0.45	1.08
Beauty and Personal Care	Text	0.31	0.49	0.63	1.46
	Text+Desc.	0.21	0.33	0.43	0.98
Cell Phones and Accessories	Text	0.24	0.44	0.50	1.41
	Text+Desc.	0.19	0.33	0.37	1.01
Electronics	Text	0.21	0.34	0.44	1.04
	Text+Desc.	0.13	0.23	0.27	0.69
Health and Household	Text	0.32	0.49	0.63	1.42
	Text+Desc.	0.18	0.31	0.36	0.93
Movies and TV	Text	1.04	1.43	1.95	3.63
	Text+Desc.	0.57	0.88	1.04	2.37
Toys and Games	Text	0.53	0.85	1.07	2.54
	Text+Desc.	0.42	0.67	0.84	1.94
Video Games	Text	0.77	1.21	1.44	3.35
	Text+Desc.	0.47	0.79	0.92	2.36

Table A.4: Comparison of Item Text Representation Methods. This table presents the results for different approaches to representing item text. ”Desc.” means Description. The scores are presented for various metrics across different categories. ”N” stands for Normalized Discounted Cumulative Gain (NDCG), and ”H” stands for Hit Rate (HR).

Appendix B

Model Configuration

B.1 SASRec

The Self-Attentive Sequential Recommendation (SASRec) [22] model was implemented and trained using the RecBole library [65, 56, 64]. It was trained for 300 epochs with a batch size of 2048, using the Adam optimizer with a learning rate of 0.001. The model architecture consists of 2 layers with 2 attention heads, a hidden size of 64, and an inner size of 256. To prevent overfitting, we applied dropout with a probability of 0.5 for both hidden and attention layers.

B.2 UniSRec(BLaIR)

We implemented UniSRec(BLaIR) using the original code provided by Hou et.al [16]. The model was configured with 2 layers and 2 attention heads, using a hidden size of 300 and inner size of 256. We set the learning rate to 0.001 and used a batch size of 2048 for both training and evaluation. Dropout probabilities of 0.5 were applied to hidden and attention layers.

B.3 Sentence-T5 (ST5)

We implemented the Sentence-T5 (ST5) models using the Sentence Transformers library [39]. The configuration includes a batch size of 64 and 10 epochs for both pretraining and fine-tuning phases. The learning rate was set to 2e-5 with a warmup ratio of 0.1.To optimize training, we employed gradient accumulation over 4 steps and enabled gradient checkpointing.

Appendix C

Data Statistics

Category	#Users	#Items	#Interactions
All_Beauty	28,570	34,547	64,557
Baby_Products	156,361	90,313	258,620
Video_Games	137,613	69,548	238,548
Beauty_and_Personal_Care	204,413	265,390	300,133
Cell_Phones_and_Accessories	183,207	228,215	275,462
Electronics	239,942	333,540	329,314
Health_and_Household	202,770	215,453	295,409
Movies_and_TV	85,832	159,537	136,205
Toys_and_Games	193,150	269,961	289,319
Books	233,260	824,855	342,060
Digital_Music	11,968	30,425	39,767
Amazon_Fashion	68,001	139,506	127,749
Yelp (2018)	231,024	143,643	503,564

Table C.1: Summary of dataset statistics across various domains