CS6964: Neuro Symbolic ML Project - Recommending Future Items based on User history

Abhishek Rajgaria
CS-6964, Neuro Symbolic ML
University of Utah
abhishek.rajgaria@utah.edu

Kushagra Dixit
CS-6964, Neuro Symbolic ML
University of Utah
kushagra.dixit@utah.edu

1 Problem Definition

Recommendation systems often face challenges in balancing accuracy, ease of understanding, and the ability to meet user preferences. This becomes particularly difficult in cases where systems need to consider user feedback and behavior patterns effectively. Traditional approaches, like collaborative filtering or purely neural methods, often struggle to capture the subtle relationships between users and products, especially when there's limited data on frequently purchased-together items. They also tend to overlook valuable context from user reviews and sentiment, which can be key to delivering truly personalized and meaningful recommendations.

This project seeks to address these challenges by using a Neuro-Symbolic Machine Learning approach combined with contrastive learning. The goal is to create recommendations that are not only more relevant but also easier to interpret. Contrastive learning helps the system understand the fine differences between items users might like and those they might want to avoid. This allows it to prioritize products with strong positive feedback and avoid those associated with negative experiences.

Our approach brings together three key ideas:

- **Contrastive Learning:** This helps the system distinguish between products based on user feedback and context, ensuring that recommendations better match user preferences.
- **Symbolic Reasoning:** By embedding clear rules, like frequently purchased-together products or user-defined constraints, the system can create recommendations that feel more logical and aligned with specific needs.
- **Neural Representations:** These allow the system to uncover hidden patterns in large datasets, like the Amazon Reviews dataset, and use this knowledge to make better recommendations.

By combining these elements, this project aims to overcome some of the common limitations of traditional systems, such as their inability to handle complex user constraints while remaining easy to understand. The inclusion of contrastive learning ensures that the recommendations are not only accurate but also closely aligned with real-world user preferences, making the system more helpful and user-friendly.

2 Motivation

Recommendation systems have become essential tools in e-commerce and digital platforms, helping users navigate the overwhelming variety of options available. However, many existing systems struggle to meet the growing demand for personalized and understandable recommendations. Traditional approaches, like collaborative filtering or deep learning models, often work behind the scenes in ways that are difficult to interpret, making it hard for users to trust the recommendations they receive. These systems also face challenges when it comes to handling specific user needs or preferences, especially when those needs involve feedback or behavioral patterns.

Modern datasets, such as the Amazon Reviews Dataset, add another layer of complexity. These datasets contain extensive product details, user reviews, and behavioral data, but often lack clear connections between related products, such as those frequently purchased together. This fragmentation can limit the effectiveness of traditional recommendation systems, which are not designed to make full use of such rich but scattered information.

The motivation for this project stems from the potential of combining symbolic reasoning, which adds clarity and structure, with the flexibility of neural networks. This blend allows us to address challenges like understanding user preferences and creating recommendations that feel more tailored and logical. Adding contrastive learning to the mix further refines the process, enabling the system to learn from both positive and negative feedback. This way, the system can focus on recommending products that align with what users truly want while avoiding those that may not meet their needs.

By embedding clear rules, such as identifying frequently co-purchased products or considering user sentiment, the system becomes not only more accurate but also easier to understand and trust. This approach is particularly useful in scenarios where users have specific requirements, like budget constraints or seasonal preferences. The motivation behind this project is to develop a framework that can address these gaps, creating a recommendation system that is smarter, more adaptable, and better aligned with the real-world needs of its users.

3 Related Work

Recommendation systems have come a long way from traditional methods like collaborative filtering to more advanced approaches that incorporate a mix of neural and symbolic reasoning. Sequential recommendation models, for example, have been widely used to analyze user behavior over time. These models are especially useful for predicting what a user might need next, based on their recent activity. They excel in time-sensitive situations, where understanding the sequence of user actions can provide valuable insights Chen et al. [2022]. However, these systems often struggle to include explicit rules, such as avoiding negatively reviewed items or adhering to specific user constraints like budget. Additionally, their heavy reliance on sequential data can make them less effective in cases where user behavior isn't time-dependent. To address these shortcomings, our approach combines symbolic reasoning such as contrastive learning, data augmentation, allowing us to better meet user-specific needs while maintaining an easy-to-understand framework.

The BLAIR framework, which utilizes the Amazon Reviews 2023 dataset, highlights the importance of pretrained embeddings for improving recommendation quality. By focusing on item metadata and user-generated content like reviews, BLAIR has shown its ability to generalize across various domains, including complex product search Hou et al. [2024]. While BLAIR offers a strong foundation for representing text and product information, our project builds on this by embedding clear association rules and symbolic reasoning directly into the system. This allows our model to provide recommendations that not only leverage textual data but also align more closely with specific user-defined constraints, leading to a more personalized and practical experience.

An important part of our work involves identifying frequently co-purchased products, inspired by the widely used Apriori algorithm Agrawal and Srikant [1994]. Apriori is a foundational tool in data mining that identifies patterns of frequently occurring items in large datasets. We used this algorithm on the Amazon Reviews 2023 dataset to uncover relationships between products that are often bought together. These patterns were then incorporated into our system to improve cross-selling opportunities and create more meaningful recommendations. Unlike purely neural methods that infer patterns indirectly, our approach explicitly uses these rules, making the recommendations easier to understand and trust.

Another key influence on our work is research on combining logical rules with machine learning models, particularly for tasks like relation extraction Rocktäschel et al. [2015]. This research demonstrated how simple logical rules can be integrated into models to improve their interpretability and ability to generalize. Inspired by this, we included logical rules in our system to guide recommendations. For instance, by prioritizing products frequently purchased together or avoiding those with negative reviews, our system can make smarter and more relevant suggestions. This combination of symbolic rules and neural models ensures that the recommendations are both practical and explainable.

In addition, we incorporated contrastive learning to fine-tune the recommendation process. This approach helps the system distinguish between products users are likely to prefer and those they might want to avoid, based on feedback and context. By focusing on this differentiation, our system ensures that the recommendations align better with what users actually need or want. This sets our work apart from traditional systems that often overlook feedback when generating recommendations.

Overall, our project brings together ideas from sequential models, pretrained embeddings, data mining, and logical reasoning to tackle some of the key challenges in recommendation systems. By combining neural adaptability with clear and interpretable logic, and refining the process through contrastive learning, we've created a framework that balances practicality, user focus, and scalability. This approach offers a step forward in creating recommendation systems that are both smarter and more aligned with real-world user needs.

4 Dataset

The study utilizes the "2023 Amazon Reviews Dataset" Hou et al. [2024], compiled by the McAuley Lab and sourced from Hugging Face. This extensive dataset spans user purchases from May 1996 to September 2023, encompassing 571.54 million user reviews and approximately 48 million products. It includes rich item metadata such as descriptions, prices, average ratings, and other attributes. Furthermore, the dataset provides detailed temporal information and user-item interaction graphs, enabling in-depth analysis of user behavior and product trends.

The dataset is organized into 33 main categories; however, several complex attributes, such as "bought together" relationships and "hierarchical categories," are not available for most categories. Additionally, the distribution of users and products varies significantly across categories, as expected. Certain categories contain a disproportionately large volume of data, leading to memory constraints during processing. For our study, we focused on user histories, ensuring that each item in the history included at least a category, user review, and rating. To address memory limitations and align with our requirements, we restricted our experiments to three major categories: "Baby Products," "Appliances," and "Handmade Products." Table 1 summarizes the dataset statistics, including the number of users, products, ratings, and review tokens for these selected categories.

Table 1: Dataset Statistic for 3 Categories

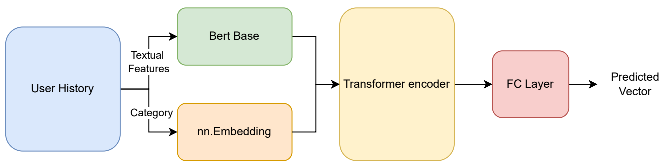
Category	#User	#Item	#Rating	#R_Token
Baby_Products	3.4M	217.7K	6.0M	323.3M
Appliances	1.8M	94.3K	2.1M	92.8M
Handmade_Products	586.6K	164.7K	664.2K	23.3M

The length of user purchase histories within the dataset exhibits substantial variation, ranging from 1 to 539 items. As our objective is to develop a model that learns user interests from their purchase histories, we included only those users whose history contained at least 20 items. Additionally, we truncated the maximum history length to 50 items to manage computational efficiency and ensure consistent input dimensions. Each history entry includes the following attributes: "user rating," "review title," "review text," "product ID," "timestamp," "product main category," "product features," "product description," "product average rating," "product title," and the "bottom category" from the product's hierarchical classification. Based on these criteria, 7,297 users and 43,528 distinct products were selected for the analysis.

Although the dataset includes a "bought together" field in the product metadata, this field is empty. To identify frequently bought together product pairs, we applied the Apriori algorithm with a minimum support threshold of 0.0005, treating a product pair as frequent if its co-occurrence exceeded 22 instances. This approach yielded a total of 717 frequent item pairs, which were subsequently utilized to augment the dataset through a logic-based approach.

5 Methods Applied

We explored various approaches discussed in class and established two baseline methods for the recommendation task.



High level model architecture for multi-label based model

- **Multi-Label Prediction/Classification Baseline**: In this approach, the recommendation task was reformulated as a multi-label prediction problem. A target vector was constructed where indices corresponding to the encoded values of products in the user history (using a product encoder implemented via sklearn's LabelEncoder) were set to 1, and all other indices were set to 0. This allowed the recommendation problem to be framed as a multi-label classification task, which was particularly advantageous as it accommodated the irregular lengths of user histories.

- **Decoder Baseline**: In this method, we adopted an decoder framework, fixing the prediction output to a maximum of 5 products based on an input user history of length 15. This ensured a consistent input-output structure, simplifying the prediction task

We incorporated techniques discussed in the class such as data augmentation, logic-based enhancements in the loss function, and architectural modifications exclusively within the multi-label baseline. This decision was guided by the flexibility and simplicity offered by the multi-label approach. While architectural changes were also attempted with the encoder-decoder baseline, they yielded no significant improvement, producing results comparable to the multi-label approach. Consequently, other class-discussed methods were not explored further for the encoder-decoder framework.

5.1 Multi-Label Baseline

In this approach, the user history is divided into two segments: an input sequence of size 15 and the remaining portion designated for future prediction. The future prediction is represented as a target vector, where a value of 1 indicates that a product is part of the future sequence, and 0 otherwise. This formulation enables the model to handle and infer on irregular user history lengths effectively.
We leveraged multiple data modalities, including user reviews, product features, product titles, and category information. User reviews, product features, and titles, being textual in nature, were processed using a pre-trained BERT model from Hugging Face. Categorical information, such as product categories, was encoded using nn.Embedding layers.
The sequential outputs from the BERT model were combined with the categorical embeddings of items in the sequence and further encoded using a Transformer Encoder. Finally, a fully connected network was applied to the aggregated encoded representations of the sequence to generate prediction logits. Sigmoid activation was employed to convert the logits into probabilities, facilitating multi-label prediction. For loss function we used Binary Cross Entropy as each vector value could be either 1 or 0 in the target vector.

5.1.1 Augmenting Data

In this approach, we augmented the training data using a first-order logic-based method that leverages pairs of products frequently purchased together. These product pairs, identified using the Apriori algorithm, were incorporated into the dataset to enhance the model's ability to learn associations between co-purchased items, thereby improving its predictive performance.
For data augmentation, we iterated through each user history during training. If a product from the history was found in the frequent pairs, its associated pair was explicitly marked as 1 in the target vector, indicating that the associated product was purchased in the future. This augmentation was implemented at the data loader level exclusively during training, ensuring that the validation and testing datasets remained unaltered to maintain the integrity of performance evaluation. The rest of the model architecture remained same.

If Product j and Product i are frequently purchased together, this relationship can be formalized using a logical implication:

bought(User x, Product j)$\rightarrow$bought(User x, Product i)

where the rule suggests that if User x has purchased Product j, it is likely that they will also purchase Product i. This rule underpins the augmentation process by leveraging the frequent co-occurrence of product pairs to enhance model training with logical inferences.

5.1.2 Contrastive Learning

In this approach, we incorporated a contrastive loss to account for the user's positive and negative reviews of products in their input history. The goal was to guide the model towards predicting products likely to receive positive reviews while avoiding products similar to those with negative reviews.

To integrate the contrastive loss, we extracted sequential embeddings for each item in the input sequence using the Transformer Encoder. These embeddings encapsulated the user review titles in the latent space. The embeddings were further processed using torch.norm to convert them into scalar values, which were treated as review scores. These computed scores were then compared against the actual user-provided ratings to measure the alignment between the predicted embeddings and the user's sentiment, enabling the model to distinguish between positively and negatively reviewed products effectively.

The contrastive loss was formulated as the sum of positive and negative losses.

- Positive Loss: This component encouraged alignment between the embeddings of products with high user ratings and their corresponding review scores. The goal was to ensure that products with higher ratings were closer in the embedding space to representations of positive sentiment.

  PositiveLoss = $\frac{1}{N}\sum_{i=1}^{N}(1-s_i)\cdot r_i$

  here N represent input sequence length, s_i represent review score and r_i represent normalized rating

- Negative Loss: This component penalized the alignment of products with low ratings, ensuring that embeddings for negatively reviewed products were pushed away from positive sentiment representations in the embedding space

  NegativeLoss = $\frac{1}{N}\sum_{i=1}^{N}\max(s_i - margin, 0)\cdot(1-r_i)$

  here margin is 0.5, and rest is same as positive loss.

  Contrastive Loss = Positive Loss + Negative Loss.

  During training, the contrastive loss was combined with the cross-entropy loss to form the final training objective. The cross-entropy loss focused on optimizing the multi-label prediction task by aligning the predicted probabilities with the ground truth target vector. Meanwhile, the contrastive loss supplemented this by encouraging the model to distinguish between positively and negatively reviewed products based on their embeddings. Rest architecture remained same as baseline.

5.1.3 Architectural changes

In this approach, we introduced a minor architectural modification inspired by an implication rule to address the large prediction space, which arises from the significant number of distinct products. To reduce this prediction space, we implemented a multi-stage prediction mechanism. Initially, the model predicts the categories of the future items, effectively narrowing the scope of potential products by identifying relevant categories.

Based on the predicted categories, we applied a masking operation to the target prediction space, where products belonging to the predicted categories were marked with 1, and those outside these categories were masked with 0. This process ensured that the model's final predictions were restricted to products within the predicted categories, thereby reducing the prediction space and improving the efficiency and potential accuracy of the recommendation process.

6 Evaluation

As the focus of our task is on predicting items irrespective of their order, we employed top-k Recall and top-k precision as evaluation metrics to assess the performance of the various approaches. These metrics effectively capture the model's ability to recommend relevant items within the top-k predictions, providing a measure of the recommendation quality. We varied value of k among 1, 3, 5

7 Experiments & Results

To encode the textual information, we utilized the BERT model `bert-base-uncased` from Hugging Face. Between the encoder and the fully connected layer, we applied Layer Normalization and GELU as the activation function to enhance training stability and non-linearity. The Adam optimizer was employed with various learning rates 1e-3, 1e-4, 1e-5to identify the optimal configuration. For training, we leveraged the On-Demand CHPC resources provided by the University of Utah, which offered access to available GPUs for efficient model training.

Table 2: Result for Top@k Recall and Precision

	Top@1		Top@3		Top@5	
	Recall	Precision	Recall	Precsion	Recall	Precision
Baseline	0.00%	0.00%	0.00%	0.00%	0.07%	0.01%
Augmented Data	3.97%	3.97%	7.81%	2.76%	11.85%	2.63%
Contrastive learning	3.97%	3.97%	9.11%	3.33%	11.71%	2.60%
Architectural change	0.00%	0.00%	0.00%	0.00%	0.07%	0.01%

Table 2 represents the results for the multi-label baseline approach, combined with various techniques discussed in class, reveal notable insights. The baseline results were suboptimal, indicating that the model struggled to make accurate predictions. Upon further analysis, it was observed that product item names and user reviews often contained very similar words. Although the model recommended relevant products, these recommendations did not exactly match the user's future purchase history, leading to poor performance metrics.

In the architectural change approach, the masking mechanism did not yield significant improvement. The model encountered difficulty in accurately predicting future product categories, which subsequently led to random or poor product predictions. Conversely, the data augmentation and contrastive learning approaches demonstrated comparatively better performance.

For the data augmentation approach, analysis revealed that the model tended to predict products that were more frequently purchased. This behavior, which aligns with the architecture's focus on frequent co-occurrences, resulted in improved recall. However, the number of predicted items also increased, leading to a decline in precision relative to recall. This observation underscores the importance of including precision in the evaluation metrics.

The contrastive learning approach showed promising results as well, with the model predicting items that had higher overall average ratings compared to the baseline predictions. Analysis further highlighted a direct correlation between high average ratings and the frequency of product purchases, which likely contributed to the improved predictions under this approach.

8 Learning & Challenges

This project has been a deeply rewarding and challenging experience, offering us a chance to explore new ideas and learn about the exciting field of neuro-symbolic learning. This approach combines the logic of symbolic reasoning with the flexibility of neural networks, giving us a way to make recommendations that are both accurate and easy to understand. We discovered how embedding clear rules such as user preferences or product patterns into models can improve their ability to provide meaningful and user-focused recommendations. This learning has expanded our understanding of what recommendation systems can achieve and inspired us to think about them in a more structured and adaptable way.

Challenges in Predicting Future Purchases: One of our main goals was to train models that could predict whether the recommended products matched what users might buy in the future. However, we faced challenges when embedding products into the model. Many products in the same category had very similar titles, making it hard for the model to distinguish between them. As a result, it often recommended products from the same category or ones that were close to the user's future purchases. This reduced the model's accuracy and highlighted how difficult it can be to handle similarities within product categories.

Issues with Transformer Models: We also experimented with a transformer model to improve predictions, but the results didn't meet our expectations. The dataset's small size and the high number of unique products caused the model to rely too heavily on frequently occurring items in user histories. This made it predict common items rather than learning meaningful patterns. The distinct nature of the products and the lack of overlap between them limited the model's ability to generalize effectively.

Trying a Hierarchical Approach: To deal with the large variety of products, we tried a step-by-step approach. First, the model predicted the category of a product, and then it narrowed down to specific products within that category. However, the dataset included about 900 subcategories, and the model often defaulted to predicting only the most common categories, skewing the results. We tried to fix this by adjusting the model's learning process to give less weight to the most popular categories, but the improvement was minimal. This showed us how challenging it is to balance predictions across a wide range of categories.

Limited Resources and Data Challenges: Resource constraints were another challenge we faced. The dataset included only a small number of users compared to the vast number of products, most of which were unique. This imbalance meant that there were very few products that users frequently purchased together, making it hard for the model to find meaningful patterns. Without enough overlapping data, it was difficult to create strong recommendations that could work for a larger audience.

Dealing with an Imbalanced Dataset The dataset itself was heavily imbalanced, with many products appearing only once or twice. This created a huge prediction space, making it challenging for the model to focus on relevant items. In recommendation tasks, this kind of imbalance often leads to a focus on popular items while ignoring less common but equally important products. The sheer number of unique products also made it harder to balance accuracy with diversity in the recommendations. This highlighted the importance of finding better ways to handle datasets with such imbalances.

9 Future Work

In our research we want to create a smarter and more adaptable recommendation system. However, there are several ways we can improve and expand upon our work to make it even more effective and user-friendly.

Using a Larger and More Diverse Dataset: Currently, our analysis is based on a dataset of approximately 7,000 users, which, while valuable, includes a high number of unique products with limited overlap. This means that we observed relatively few patterns of products being frequently purchased together. This limitation impacted our ability to identify strong co-purchase trends that could improve recommendation quality. In the future, we aim to include data from more categories and a larger number of users. By expanding the dataset, we can uncover richer patterns and relationships between products, leading to more accurate and reliable recommendations. This broader dataset could also help us cater to a more diverse range of user preferences and behaviors.

Exploring More Models: While we experimented with sequential models like transformers, we recognize the need to evaluate other models that might better suit specific aspects of the recommendation process. For example, Graph Neural Networks (GNNs) could allow us to map relationships between users and products more effectively, as they are designed to understand connections in a network. Additionally, Field-Aware Factorization Machines (FFMs) could be valuable for handling the detailed, multi-field information in our dataset, such as user preferences, product categories, and review sentiments. By comparing these models with our current approach, we could identify methods that offer improved accuracy, better handling of diverse data types, and greater adaptability to varying user needs.

Adding Timestamp Information to Recommendations: Our current system focuses on product titles and reviews to generate recommendations. However, the dataset includes timestamps that indicate when purchases were made, a valuable piece of information we have not yet used. Including these timestamps could enable the system to make more contextually relevant recommendations. For instance, it could identify seasonal trends, such as recommending holiday decorations during the winter or school supplies in late summer. Additionally, using timestamps would allow us to analyze user purchase histories over time, helping us better understand how preferences change and make recommendations that align with those evolving patterns. This would create a system that feels more responsive and aligned with user needs.

Understanding User Similarities: At present, our system treats each user as an independent individual, without exploring how users may be connected by similar preferences or behaviors. This approach misses an important opportunity to use shared patterns to predict behavior. For example, if two users have similar purchase histories, a product enjoyed by one user might be highly relevant to the other. In the future, we want to identify these similarities and use them to improve recommendations. By grouping similar users, we can better predict their preferences and provide more personalized suggestions. This approach would enhance the system's ability to make recommendations that feel tailored and intuitive.

References

Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, page 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1558601538.

Xin Chen, Alex Reibman, and Sanjay Arora. Sequential recommendation model for next purchase prediction, 07 2022.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation, 2024. URL `https://arxiv.org/abs/2403.03952`.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In Rada Mihalcea, Joyce Chai, and Anoop Sarkar, editors, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1118. URL `https://aclanthology.org/N15-1118`.