An Amazon Recommender using Two-Tower Recall

Rachel Chang Yutong Liu Yi-Jyun Sun Avaneesh Kumar

1 INTRODUCTION

We propose to build an Amazon-style recommender centered on a Two-Tower (user/item) representation model. The system learns user and product embeddings from reviews and product metadata to enable candidate recall, and presents results in an interactive web demo. Specifically, entering a `user_id` will return a personalized Top-K list with image, title, price, discount, and a brief recommendation reason.

2 SUMMARY

2.1 Software to be Implemented

The proposed system consists of three major components:

(1) **Data preparation and feature construction.** Clean the Amazon dataset, extract key attributes, and build composite item representations and user interaction histories.
(2) **Two-Tower based candidate recall.** Train a dual-encoder model to align user and item embeddings, enabling efficient Top-K recommendation. Optionally, provide human-readable rationales.
(3) **Interactive web interface.** Develop a lightweight demo where entering a `user_id` returns a ranked Top-K list with image, title, price, discount, and a brief reason.

2.2 Data

We use the UCSD Amazon Review dataset (2023)[1], which contains 571.54M reviews on 48.19M items from 54.51M users. Compared to earlier versions, it is larger, more up-to-date, and directly relevant to our project. Reviews are parsed into rich product metadata in JSON, including features, prices, and multimodal fields (images, videos). The dataset also provides millisecond-level timestamps, enabling precise temporal analysis.

2.3 Methods to evaluate the software

For evaluation, we use a Leave-K-Out (LKO) split per user: the last K interactions are held out for test, the second last K for validation, and the rest for training. This temporal protocol reflects realistic user behavior and ensures forward-looking evaluation.

We report two complementary metrics:

- **Hit Rate@K (HR@K).** Indicates whether at least one of a user's held-out items appears in the Top-K list: HR@K $= \frac{1}{|U|}\sum_{u\in U} \mathbf{1}\{R_u^K \cap T_u \neq \emptyset\}$, where R_u^K is the Top-K recommendations for user u and T_u the held-out set.
- **Recall@K.** Measures the fraction of held-out items that are recommended, Recall@K $= \frac{1}{|U|}\sum_{u\in U} \frac{|R_u^K \cap T_u|}{|T_u|}$.

2.4 The timeline & Activities per team member

Activities per team member.

Person A (Yutong Liu). Responsible for data preprocessing: deduplication, normalization, time series training, splitting of train, validation and test sets, negative sampling, and ID mapping. Also responsible for the use of the `load_dataset()` function and the writing of reports related to the dataset.

Person B (Yi-Jyun Sun). In charge of feature construction, such as building product text fields, calculating SBERT/E5 vectors, and preparing product metadata. Additionally, this includes adding numerical features such as price, discount, and rating counts, and performing scaling.

Person C (Rachel Chang). Responsible for model implementation and evaluation, establishing a baseline Faiss search, training the TwinTower model using ID, text, and numerical features, and obtaining product vectors. Building an IVF+PQ index and providing evaluation scripts.

Person D (Avaneesh Kumar). Developing front-end and back-end, implementing product card UI, search functionality, and delivery system demonstrations.

Timeline.

Period	Deliverables	Resp.
Sep 9–30	Cleaned data; splits; ID maps; loader	A: data/splits B: emb
Oct 1–16	Baseline retrieval; HR/Recall; report	A: integrity B: prep C: baseline
Oct 17–Nov 15	Two-Tower; item vecs; Faiss IVF+PQ	B: num. feats C: train D: API
Nov 16–30	Pipeline; demo; draft report	A: verify C: artifacts D: demo
Dec 1–9	Results; demo; report; slides	A: rpt (data) B: rpt (feats) C: rpt (model) D: demo/slides

Table 1: Condensed project timeline with deliverables and responsibilities.

[1]https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews