Graph Meets LLM for Review Personalization based on User Votes

Sharon Hirsch
Ben-Gurion University of the Negev
Beer-Sheva, Israel
eBay Research, Netanya, Israel
hirschsh@post.bgu.ac.il

Lilach Zitnitski
Ben-Gurion University of the Negev
Beer-Sheva, Israel
lilachzi@post.bgu.ac.il

Slava Novgorodov
Tel Aviv University
Tel Aviv, Israel
slavanov@post.tau.ac.il

Ido Guy
Ben-Gurion University of the Negev
Beer-Sheva, Israel
gid@bgu.ac.il

Bracha Shapira
Ben-Gurion University of the Negev
Beer-Sheva, Israel
bshapira@bgu.ac.il

Abstract

Review personalization aims at presenting the most relevant reviews of a product according to the preferences of the individual user. Existing studies of review personalization use the reviews authored by the user as a proxy for their preferences, and henceforth as a means for learning and evaluating personalization quality. In this work, we suggest using review votes rather than authorship for personalization. We propose *MAGLLM*, an approach that leverages heterogeneous graphs for modeling the relationships among reviews, products, and users, with large language model (LLM) to enrich user representation on the graph. Our evaluation over a unique public dataset that includes user voting information indicates that the vote signal yields substantially higher personalization performance across a variety of recommendation methods and e-commerce domains. It also indicates that our graph-LLM approach outperforms comparative baselines and algorithmic alternatives. We conclude with concrete recommendations for e-commerce platforms seeking to enhance their review personalization experience.

CCS Concepts

• **Information systems → Online shopping**; **Personalization**.

Keywords

Recommender Systems; E-commerce; Product Reviews; Large Language Models

ACM Reference Format:
Sharon Hirsch, Lilach Zitnitski, Slava Novgorodov, Ido Guy, and Bracha Shapira. 2025. Graph Meets LLM for Review Personalization based on User Votes. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, Sidney, NSW, Australia, 11 pages. https://doi.org/10.1145/3696410.3714691

1 Introduction

Online reviews play a central role in the success of e-commerce platforms, allowing potential buyers to gain insights about a product from customers who have already purchased it. With their growing popularity, many products accumulate a large number of reviews, making it impractical for potential buyers to traverse all of them. Review personalization aims at surfacing the most relevant reviews for each user, based on their own characteristics and preferences.

Most works on review personalization define the task as a recommendation task, where given a product and a user, the goal is to recommend the top k product reviews that would be most helpful for the user [11, 20, 21, 36]. The evaluation of such a task is nontrivial, since it requires feedback from specific users about the reviews they prefer. The common approach within the review personalization literature is to leverage review authorship to gain ground truth information about user review preferences [11, 20, 39]. The underlying assumption is that reviews produced by a user are reflective of the reviews they would like to consume. The information about user-review authorship is available in many public review datasets (e.g., [35, 48]) and relying on it as a proxy for user preferences can serve as a basis for both user modeling and for creating a test set where success is defined as recommending to the user their own authored review [20].

In this work, we argue that relying on authored reviews for the review personalization task has two fundamental drawbacks. First, the underlying assumption that the content produced by users is similar to the content they would prefer to consume, is questionable, as has been shown in previous work [16, 17]. It is preferable to rely on information that reflects user preferences in terms of consuming product reviews as a more direct signal. Second, review authorship provides a sparse signal. As in many other social systems, the majority of users are lurkers; that is, they consume reviews but do not produce them.

To overcome these two shortcomings, we suggest leveraging a different signal that associates users with reviews, reflecting their preferences for reviews they *consume*. To this end, we observe that many e-commerce platforms give users the opportunity to provide explicit feedback on reviews written by other users. This type of feedback is orthogonal to the feedback (ratings) users can provide for the product itself. The screenshots of this functionality across four popular e-commerce platforms are demonstrated in Appendix A. There are subtle differences between these platforms. For instance, review feedback can be a simple "helpful" indicator (e.g., Amazon, Aliexpress), a thumbs up or thumbs down for "liking" or "disliking" (Walmart), or multi-dimensional with "helpful", "thanks", "love this", and "oh no" (Yelp). The effort required to vote on a review is substantially lower than review authorship. In our analysis, we

WWW '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1274-6/25/04
https://doi.org/10.1145/3696410.3714691

show that not only is voting already a more frequent signal than authorship, but it also engages a considerably higher portion of the users, many of whom are lurkers, who have never been engaged in writing a review. To the best of our knowledge, no previous work has comprehensively studied review personalization based on voting information.

One of the reasons likely to withhold past work from leveraging the voting signal, is merely its absence from the vast majority of popular review datasets[1] [35, 48]. To evaluate our suggested approach, we use a public dataset that includes user-review associations by voting within an e-commerce platform, over a variety of domains. We compare the use of review authorship and review voting signals for personalization across popular recommendation methods and five different e-commerce domains. Our results indicate that using the voting signal consistently yields substantially higher personalization performance. In some cases, combining the authorship and voting signals yields additional improvements over using the voting signal alone. We note that despite its exclusion from most public datasets, many leading e-commerce platforms already enable voting functionality and naturally have access to this type of information for review personalization.

In addition to inspecting existing personalization approaches, we suggest a new personalization method, which we term *MAGLLM*. This method employs heterogeneous graph modeling to capture the relationships among products, reviews, and users. To enhance user representations, a large language model (LLM) is utilized to summarize associated reviews. In our experiments, *MAGLLM* consistently shows higher performance results over a variety of other common recommendation techniques. Similarly to the other methods, the use of the voting signal is evidently preferable to using the authorship signal to model user-review relationships in *MAGLLM*. The principal contribution of this work is twofold:

- Advocating the use of voting, rather than authorship, for review personalization learning and evaluation, demonstrating its merits through data analysis, and showing it is substantially more effective over a variety of recommendation methods.
- Suggesting a review personalization approach that models product-review-user relationships using heterogeneous graph meta-paths, with enhanced user representation using LLM, and showing its superiority over a variety of other recommendation methods across multiple e-commerce domains. The implementation of our model is publicly available at https://github.com/lilahz/MAGLLM

2 Related Work

Personalized review recommendation models consider users' past interactions and preferences expressed directly or indirectly through review engagement, enabling a tailored review reading experience. Some works (e.g., [33, 34]) attempt to predict a personalized helpfulness score by integrating the connections between the review author, review text, review reader, and the product itself into probabilistic factorization models. However, even though the mentioned studies predict helpfulness of a review (tailored to users), they only consider predefined user types, such as amateurs and experts, and do not consider individual users' preferences. Another line of research considers the different kinds of interactions that can be derived from reviews. For example, the relationships between the users reading the reviews and the users who wrote the reviews (e.g., [7]) or review-user-item relationships. For example, Wang et al. [44] use transformers for computing personalized embeddings that are later used as an input for auxiliary tasks of user-item interaction modeling and review quality modeling. Peddireddy [36] used recent shopping history and previous reviews as additional user information perspectives for review recommendation; however, as this kind of information is generally not available, the author constructed user profiles by randomly generating purchase histories and review engagements.

Aside from exploiting connections between users, items, and reviews, some works also incorporated the content of reviews. By analyzing the text of the written review, they were able to identify specific attributes and preferences, which were used for recommendation. Suresh et al. [39] extracted the product attributes with their sentiments from each review. A user profile was then formed based on the user's preferences for specific products and their qualities. These profiles were used, along with social networks information, to identify similar users. However, social network information is often sparse or completely unavailable to e-commerce platforms. Dash et al. [11] used the reviews to extract product attributes using Latent Dirichlet Allocation and later grouped similar users by preferences that were calculated using sentiment analysis. After grouping the similar users, review recommendations were calculated per group. In this case, personalization might have a smaller impact on a particular user. Huang et al. [20] used sentiment-based recommendations. They evaluate similarity between users based on the attributes and sentiments they share and use it for personalized review recommendations. Concretely, the method identifies users' aspect preferences from the reviews they wrote, calculates similarity between users with shared preferences for the same products, and ranks reviews using a helpfulness score. The written reviews are considered as a ground truth. Most of the above-mentioned works solely rely on authored reviews for review personalization, which holds its limitations as described in the previous section.

Leveraging graphs is also an established approach in other recommendation scenarios [24, 27, 38, 40]. For example, Tan et al. [40] modeled the user-item rating data and additional features using a heterogeneous graph and manually construct the meta-paths based on domain knowledge. This graph is used to extract user and item embeddings and fed as an input to a deep-learning model for rating prediction. These works often leverage user and item embeddings that were learned from the graph for rating prediction, however they mostly focus on items rather than their reviews.

3 Datasets

In this section, we first present the primary dataset used for our analysis and evaluation. We then introduce an additional dataset, employed mainly for gaining deeper insights into user behavior, particularly on how users express their preferences on reviews. Both datasets are significant to our work because they contain valuable and non-trivial information about the feedback users provide on products and reviews, which is not commonly available in other public datasets used in previous studies.

[1] Some of the datasets include information about the total votes per review, but not the individual voters.

Ciao Dataset. [41]. This is the main dataset used for the evaluation of our proposed method, and the comparison of review authorship versus review voting signal for personalization across various recommendation methods examined in our research. Ciao is a public dataset of product ratings and reviews from an European-based online-shopping portal. Ciao uses a 6-point product rating scale ranging from 0 to 5. It also allows users to express their feedback about the helpfulness of a review using numeric quality ratings ranging the same 6-point scale. We refer to this feedback hereafter as a *vote*. The dataset contains, for each review, the user identifier, product name, product category, product rating score given by the reviewer, overall review helpfulness score , date the review was written, content of the review, and a list of helpfulness voting scores associated with specific user ids.

The data was crawled from the site along the month of May 2011 and consists of 27 categories. Tables 1 and 2 present the statistics of the dataset. In total, there are 270,126 reviews, over 10,700 users who write reviews and over 42,000 users who vote for reviews. We can see that over 95% of the users interact with the products by voting for reviews, while only 22.7% of the users write reviews on products. These numbers attest to the benefit of using votes rather than authorship as basis for review personalization. We opted to use this dataset since it shares, for each user, the reviews they voted for (and the rating of the vote), while other datasets, only include the total number of votes per reviews (e.g., Amazon [35] and Yelp [48]) or do not include review vote information at all, even though the corresponding platforms do feature a helpfulness score per review (e.g. TripAdvisor and IMDB [30]). As mentioned in Section 1, this information is typically available to e-commerce sites who seek to personalize product reviews.

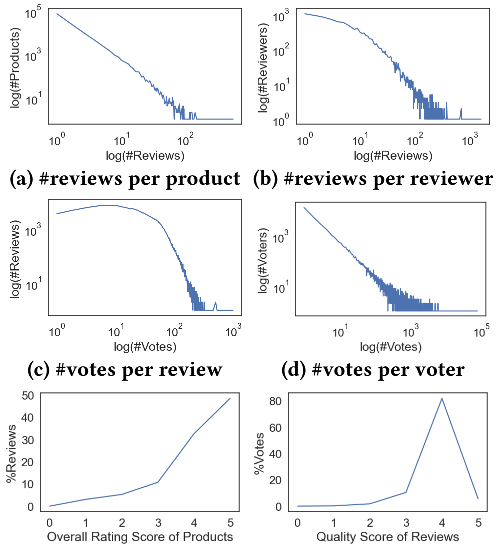
Figure 1 shows different distributions of reviews and votes in the dataset. In the first two figures, 1a and 1b, we can see the distributions of number of reviews per product and number of reviews per reviewer, respectively. Both distributions follow a power-law distribution. A large portion of products receives a small number of reviews, while a small portion of products has many reviews. As for users, few users write a large number of reviews while many users write only few reviews. A similar trend can be seen also in Figure 1d, which shows the distribution of number of votes per voter. Figure 1c shows the distribution of number of votes per review, which deviates to some extent from a perfect power-law distribution since the first values, which refer to reviews with five or fewer votes, are not the most frequent in the dataset. In addition, we examined two distributions that refer to the review rating scores. First, the distribution of overall rating scores that were given by the writer of the review (the reviewer) to the product (Figure 1e). The rating score reflects the extent to which the reviewer is satisfied with the product. It can be seen that nearly 50% give the highest score of 5, additional 32.5% give 4, and only 19% give the four lower scores of 0-3. Overall this reflects the positivity of product rating, as has also been show in past work [26]. Second, the distribution of the quality ratings of reviews (Figure 1f), given by the reader of the review (the voter). The quality score expresses how much the review was helpful for the reader. Overwhelmingly, the most common score is the second highest - 4 - at nearly 80%. 3 and 5 account for roughly 11% and 6%, respectively, and only 2.2% of the voters give the three lowest scores of 0-2.

Table 1: Ciao Dataset - Basic Characteristics.

#Products	#Reviews	#Reviewers	#Votes	#Voters
109,451	270,126	10,731	7,788,175	42,035

Table 2: Ciao Dataset - Review Writing and Voting.

Action	#Users	%Users
Write	10,731	24.26%
Vote	42,035	95.02%
Write & Vote	10,050	22.72%
Write & Not Vote	681	1.54%
Vote & Not Write	31,985	72.30%



(a) #reviews per product **(b) #reviews per reviewer**

(c) #votes per review **(d) #votes per voter**

(e) Distribution of product rating **(f) Distribution of review quality rating**

Figure 1: Ciao Dataset Analysis

Edmunds dataset. Our second dataset is also public and contains car reviews [14] collected from American Automotive online shopping site Edmunds. The total number of reviews is 42,288. Each review includes the date, the reviewer name, the full review text, and an additional 'favorite' field (manually filled by the reviewer), which highlights attributes or characteristics of the product that the reviewer especially preferred. After removing reviews with missing data, we were left with a total of 40,925 reviews. This dataset is unique because it includes an explicit signal indicating users' favorite characteristics of the product, captured through the 'favorite' text field. Leveraging this information allows us to assess how well written reviews capture user interests in the product by comparing the full review text with the additional 'favorite' field for each user.

4 Voting vs. Writing

Most of the existing works on review personalization rely on modeling approaches that utilize reviews written by users themselves to create personalized review recommendations and perform evaluation [11, 20, 36, 39, 43]. Relying only on reviews written by users has two significant limitations. First, written reviews may not capture the full spectrum of user preferences and opinions. The assumption that the reviews a user writes represent, in terms of content and

Table 3: Reviews vs. Votes Statistics across Datasets.

Dataset	Category	#Reviews	#Votes	Ratio
Amazon	Books	29,475,453	52,381,607	1.78
Amazon	CDs and Vinyl	4,827,273	9,212,281	1.91
Amazon	Camera & Photo	4,340,159	7,020,382	1.62
Amazon	Exercise & Fitness	3,193,115	5,155,488	1.61
Amazon	Games	1,502,718	2,573,243	1.71
Amazon	Hair Extensions, Wigs & Accessories	985,065	1,648,990	1.67
Yelp		6,990,280	14,048,967	2.01

style, the reviews they are also going to prefer as consumers, has never been put to test. Second, as in other user-generated content, the majority of users who consume reviews do not produce ones. Table 2 presents the statistics of review writing and voting in the Ciao dataset described in Section 3. While only 24.3% of the users write reviews, over 95% of the users vote for reviews. Similar trends can also be observed in other review datasets that are widely used in recommendation research, such as Amazon [35] and Yelp [48].

Since the information of user-review interactions is not available in these datasets (only the total amount of votes per review), we cannot calculate directly the portion of users that write or vote. Instead, for each dataset, we calculated the total number of reviews across all products to represent authorship, and the total number of votes across all reviews to represent user voting. Table 3 includes the ratio column, which represents the number of votes divided by the number of reviews. As shown in the table, in the Amazon dataset, across several popular categories, voting is substantially more prevalent than writing a review, with the number of votes exceeding the number of reviews by a factor of at least 1.6. Notably, in the CDs and Vinyl category, the ratio is nearly double. In the Yelp dataset, the number of votes (which includes three different types: useful, funny, and cool) is over double the number of reviews.

4.1 Do Written Reviews Reflect Preferences?

To validate the assumption that written reviews do not fully represent user preferences, we analyze the cars reviews dataset - the Edmunds dataset- since in addition to the reviews written by users, it includes an explicit field mentioning the favorite characteristics of the reviewed car by the user. We set out to examine the intersection between the attributes mentioned in the review and the attributes mentioned by the user in the "favorite" field. We used a neural named-entity recognition tool [6], which is currently state-of-the-art for e-commerce attribute extraction, to identify a list of car attributes from the review texts. To further expand the attribute list we used, in addition, two publicly available datasets of car attributes that were manually collected and annotated from various websites with reviews [10, 31]. We also added plural forms of singular attributes to the list and vice versa to include different variants of the attributes. Finally, we calculated how many attributes are mentioned in both the review and the "favorite" field for each user.

The analysis reveals that the average number of attributes mentioned in the review, 8.451, is almost double the number mentioned in the "favorite" field, which is 4.879. Moreover, the intersection of the two lists is quite low, on average 0.791 (i.e., less than one overlapping attribute), hence most of the attributes are different. Since the "favorite" field contains explicit user preferences, this indicates a clear gap between the content of written reviews and user preferences. Thus, written reviews do not fully represent the user preferences and relying exclusively on them may result in missing important information.

Table 4: Ciao 4-core Dataset Statistics across Top 5 Categories.

Category	#Products	#Reviews	#Reviewers	#Votes	#Voters
DVDs	2,640	27,964	3,703	602,291	5,189
Beauty	1,813	14,091	2,747	333,407	4,417
Food & Drink	1,318	12,084	2,533	349,302	4,410
Internet	790	11,737	3,079	226,115	4,965
Games	909	8,919	2,486	110,424	4,269

4.2 Using Voting Signals For User Preferences

Since written reviews are sparse and noisy and may not fully represent the user preferences, we suggest to utilize the more frequent and direct vote signal. To our knowledge, we are the first to explore the use of the *vote* signal as a primary source for capturing user preferences in review recommendation. By leveraging the data from the user-review interactions, we aim to create a more accurate representation of user preferences and improve the personalized review recommendation.

To assess the effectiveness of harnessing the vote signal compared to using the reviews written by users, we use a dataset that includes explicit user feedback on reviews, where users both authored their own reviews and voted on reviews written by others. For the training and evaluation of our approach we create a dataset based on the Ciao dataset, which we refer to as the "4-core" dataset. Concretely, we consider all users who wrote at least 4 reviews and all products that have at least 4 written reviews, with 4 chosen based on empirical testing. The portion of the reviews we consider this way is 50% of all the reviews. Note that for products with fewer reviews, personalization is not acutely required, since the user can simply traverse all the reviews (3 or fewer). The "4-core" dataset allows to compare two different approaches on the same set of users: *write* versus *vote*. The first approach relies only on reviews written by the user, while the second approach relies only on reviews the user has voted for. Additionally, we test a third approach, a hybrid version of *both* write and vote, to examine whether the combination of these two approaches provides any improvement over the single-signal approaches. In our work, we focus on the five largest categories based on the number of reviews: DVDs, Beauty, Food & Drink, Internet, and Games. Table 4 presents the statistics of the categories in the 4-core dataset.

We use the 4-core dataset to compare three different strategies for personalized review recommendation: 1) using only the *vote* signal, 2) using only the *write* signal, and 3) combining *both* signals. For each signal$\in\{vote, write, both\}$, the user is associated with a different review set. Specifically, the reviews the user voted for in the *vote* signal, the reviews the user authored in the *write* signal, and a union of the two in the *both* signal. Accordingly, we define this set as the *signal-based review set* associated with a user, and refer to this definition throughout our experiment and result description. We consider reviews with scores of 3, 4, or 5 as positive or "liked" reviews, and reviews with scores of 0, 1, or 2 as negative or "disliked" reviews. When using the *vote* signal for learning user preferences, we consider only positive reviews in the *signal-based review set*. We examine each of the signals across a variety of recommendation methods, including our proposed approach based on heterogeneous graph modeling and LLM-generated user profiles.

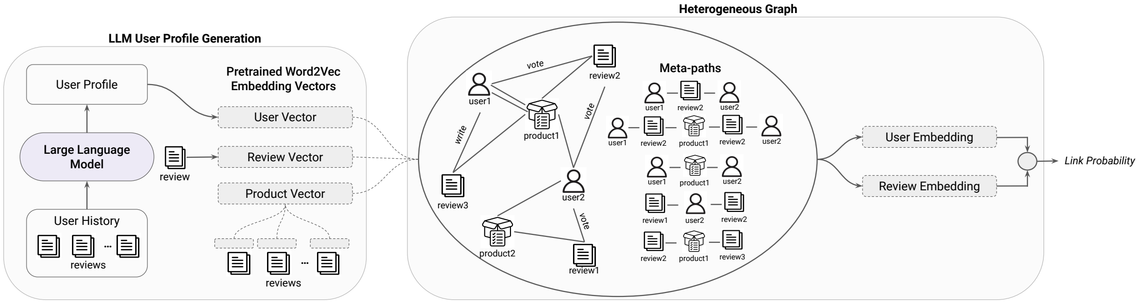


Figure 2: An overview of our proposed method *MAGLLM* for personalized review recommendation

5 Our Suggested Approach

We propose *MAGLLM*, a personalized review recommendation technique that models relationships among products, reviews, and users using a heterogeneous graph network with meta-paths and enhances user representation using LLM. Our approach consists of two components: (1) a user profile generated using LLM to provide a rich representation of users based on the reviews they wrote and/or voted for, and (2) a heterogeneous graph-based model with meta-paths originally designed for the link prediction task and adjusted for the review personalization task. The graph structure is aimed at capturing the diverse and complex relationships between users, reviews, and products. The predicted link connects a user to a review, with the score reflecting the probability that the user will like the review.

An overview of the *MAGLLM* architecture is illustrated in Figure 2. First, the user’s review history is provided as input to the LLM, which generates a user profile summary in natural language that serves as the user’s representation in the graph. To convert the profile into user node features, we compute the average of the Word2Vec embeddings [32] of its words. For Word2Vec embeddings, we use the CBOW model with negative sampling, a vector size of 300, and a window size of 8. The model is trained on two review corpora, Ciao and Amazon, which include multiple categories. We opted to use an unweighted average of the embeddings after experimenting with a weighted average using TF-IDF, which showed slightly lower performance. Similarly, the review and product node features are represented by Word2Vec embeddings. A review vector is represented by the average of its words, while a product vector is derived by averaging all its review vectors. Throughout this work, we use the average Word2Vec embedding approach detailed above to represent any review or set of reviews. A heterogeneous graph is constructed using the users, reviews, and products as nodes, along with the connections (edges) between them and the predefined meta-paths . Finally, by analyzing the learned embeddings of user and review nodes, we perform link prediction to estimate the probability that a user will like a specific review.

User Profile Generation. To comprehensively capture user preferences and topics of interest, our goal is to generate a high-level summary of user preferences, incorporating as many relevant details as possible. To this end, we leverage an LLM to generate a summary by analyzing the history of reviews a user has interacted with, whether through writing or voting. This allows the LLM to extract valuable insights into the user’s preferences, such as favored products or sentiment toward specific attributes. The user profile is built independently for each category and signal by considering the reviews in the *signal-based review set* linked to the user across all other categories. From this list, we randomly sample 5 reviews to create the history of the user and truncate each review to the first 150 tokens due to LLM prompt size limitations. The summary produced by the LLM is then converted into a user vector using Word2Vec embeddings. We experiment with several prompt formats using different content and style, and present here the one that achieved the best performance (additional prompts are provided in Appendix B.1). The LLM prompt is structured as follows:

```
You are asked to describe user interests and preferences based
on his/her {signal} reviews list, your'e given the user's past
{signal} reviews in the format:
<product category, product title> : <product review content>
You can only reply the user interests and preferences (at most
10 sentences). Don't use lists, use summary structure. The
output should begin with the word Profile. These are the
{signal} reviews:
<product-1-category, product-1-title>: <review-1-content>
…
<product-5-category, product-5-title>:<review-5-content>
```

where the {signal} token is replaced with ‘voted’ or ‘written’ or ‘voted or written’ according to the specific signal. We use LLaMA-7B model [42] with the implementation of llama-cpp-python [2] as our LLM.

We explore various alternatives for tuning prompt parameters to generate user profiles, elaborated in Appendix B.2. We also explore an alternative strategy for leveraging LLMs to our task, which was not found productive, as described in Appendix B.3.

Heterogeneous Graph Modeling. Unlike traditional graphs, where nodes and edges are of a single type, heterogeneous graphs allow for the representation of multiple types of nodes (e.g., users, products, reviews) and edges (e.g., a user writing a review, or a user rating a product). Meta-paths are a set of relationships, which represents sequences of specific node and edge types. Our problem involves diverse relationships between users, reviews, and products, which requires a structure capable of modeling complex data interactions that capture rich and diverse information. A heterogeneous graph combined with meta-paths provides the necessary framework to achieve this, which is why we chose this approach for our model. Specifically, we employ the MAGNN [13] implementation [1], which uses aggregation over meta-paths to incorporate information not only from the two endpoints, but also from intermediate nodes along the path, ultimately generating node embeddings. We “transformed” the link prediction task into a review personalization task by using the prediction score for a link between a user node and a review node as a point-wise indication of the preference of

Table 5: Graph statistics across signals.

Signal	Nodes	Edges	Meta-paths
Vote	# Voter (V): 5,959 # Review (R): 74,795 # Product (P): 7,470	# V-R: 951,730 # R-P: 74,795 # V-P: 653,043	VRV, VRPRV, VPV, RPR, RVR, RPVPR
Write	# Author (A): 5,672 # Review (R): 74,795 # Product (P): 7,470	# A-R: 74,795 # R-P: 74,795 # A-P: 74,438	ARPRA, APA, RAR, RPR, RPAPA
Both	# Voter (V): 5,959 # Author (A): 5,672 # Review (R): 74,795 # Product (P): 7,470	# V-R: 951,730 # A-R: 74,795 # R-P: 74,795 # V-P: 653,043 # A-P: 74,438	VRV, VRPRV, VPV, ARPRA, APA, RVR, RAR, RPR, RPVPR, RPAPR

the specific user towards the specific review. The personalization task can then be performed by ranking all the reviews of a product based on their link prediction score to the user in question. The link prediction score, which represents the strength of the relation between the user and the review, is computed by applying a sigmoid function to the dot product of the user and review embeddings derived from the graph.

For each signal$\in\{vote, write, both\}$, we constructed a heterogeneous graph with meta-paths to describe the user-review, user-product and review-product interactions. Connections (edges) between a user and a review or product were created according to the *signal-based review set* associated with a user. For instance, for signal=$vote$, a user will be connected to reviews she voted for and the products associated with these reviews, but not to reviews she authored. Similarly to Fu et al. [13], we present the graph statistics of the signals in Table 5. The table shows for each signal the number of nodes and edges in the graph structure, and the meta-paths. The meta-paths were manually created based on domain knowledge and include only paths that start or end with users and reviews, as our focus is on personalizing reviews for users. The graphs consist of multiple node types including Voter (V), Author (A), Review (R), and Product (P). Using meta-paths allows the formation of complex relationships between nodes. For example, the meta-path *Voter-Review-Voter* (VRV) represent a connection between two different users who voted for the same review, and the meta-path *Author-Review-Product-Review-Author* (ARPRA) represent two users who wrote reviews for the same product. In addition, we use node content features for products, reviews, and users.

6 Evaluation

In this section, we describe the models and metrics used in our evaluation. Our goal is twofold: first, to compare the effectiveness of the voting signal (*vote*) versus the authorship signal (*write*) for personalization across various recommendation methods, from basic to more advanced ones. Second, to compare these methods with our proposed approach and show that it outperforms them all.

6.1 Recommendation Methods

6.1.1 Non-personalized baselines. In this group, we consider two basic methods that do not personalize the recommended reviews.

- **Random** - randomly sort the review list.
- **Popularity** - recommend the most popular reviews to all users. Popularity is calculated by the number of positive votes (equal or greater than 3). While simple, popularity is known to often produce a strong baseline [22].

6.1.2 Content-based methods. In this group of models, we create a user profile based on the content of their associated reviews in the *signal-based review set.* We represent a review using an embedding vector, and the user profile and product are represented accordingly by averaging their corresponding review vectors. The reviews of a product are ranked based on their cosine similarity score with the user profile.

- **Word2Vec** - learn vector representations of words and capture their semantic relationships. The user profile is calculated by averaging all review vectors in the user's *signal-based review set.*
- **Top Terms** - extract the top k frequent terms (excluding stop words [28]) from the *signal-based review set* associated with a user to create a user profile. The profile is calculated by averaging the word embeddings of these top terms. Each review is also represented by the average word embeddings of its top k frequent terms. We experimented with different values of k and settled on k=20 as it yielded the best performance.
- **Sentence-BERT** [37] - learn vector representations of sentences and capture their semantic meaning for tasks like similarity and clustering. Its backbone model, BERT [12], generates contextual embeddings where a word's representation changes based on its surrounding words in the sentence. Using this model, each review is represented by the average of its sentence embedding vectors. Then, the user profile is calculated by averaging the review vectors of all reviews in the user's *signal-based review set.* We use NLTK [28] to split the review into sentences, and applied on them the pre-trained Sentence-BERT model `all-MiniLM-L6-v2`.

6.1.3 KNN Collaborative Filtering. These methods use traditional collaborative filtering (CF) based on k-nearest-neighbors (KNN) to harness user-item relationships for recommendation. We implement the models with the Surprise library [3], with hyper-parameter tuning including the minimum common k, the maximum k neighbors, and the minimum support similarity.

- **KNN item-based** [45] - a CF approach based on KNN with item-to-item similarity, where reviews are considered as the items. We built a user-review matrix based on the signal that was tested: for the *vote* signal, the entries are represented by the users votes for the reviews (rating score on a scale of 0 to 5). For the *write* signal, the entries are represented in a binary form (1 if the user authored the review, 0 otherwise). For the *both* signal, we convert the votes from numeric to binary: votes in the range of 0-2 were converted into 0, whereas votes in the range of 3-5 were converted into 1. Then, the *vote* and the *write* signals are combined using a union operation. We use cosine similarity between review vectors to find reviews similar to those the user has written and/or voted for. Then, the predicted quality rating score between a user and a specific review is calculated by a weighted average of the quality rating scores of the k-nearest reviews, with weights based on their similarity to the user.
- **KNN user-based** [18] - a CF approach similar to KNN item-based, but here to find nearest neighbors the cosine similarity is calculated between user vectors.

6.1.4 Deep learning methods. These methods leverage neural networks to model complex patterns and interactions for recommendation tasks, capturing non-linear relationships and semantic features.

- **DeepFM** [15] - a hybrid model that combines factorization machines (FM) and deep neural networks (DNN) for recommendation tasks. The FM component captures low-order interactions, while the DNN component models complex, high-order interactions among features. This is a state-of-the-art algorithm for solving binary classification problems like click prediction. We used the implementation of DeepCTR [5], with a regression task and Adam optimizer. Hyper-parameter tuning included the embedding dimension, number of hidden units, L2 regularization, and dropout rate.
- **NRMS** [47] - a news recommendation method that utilizes multi-head self-attention networks together with additive attention to learn representations of users and news. The news representations are based on the article titles and the user representations on users' browsing history. For our experiments, we used the news encoder as a review encoder that gets as input the review text with a maximum length of 350 tokens. The user encoder used a "history" of up to 10 reviews, in our case the history refers to reviews the user interacted with based on the user's *signal-based review set.* The model was trained using a negative sampling technique: for each review the user interacted with (regarded as a positive sample), we randomly sampled k reviews the user did not vote for or did not author. Our implementation is based on NRMS-Pytorch [4] with k=3. We use Adam optimizer and performed hyper-parameter tuning for the learning rate, weight decay, batch size, and dropout rate.

6.2 Dataset Preparation

The ground truth for the ranking of product reviews for each user was determined by the actual rating scores (from 0 to 5) they assigned to the reviews they voted for. The reviews were sorted in descending order based on these scores. Reviews with no votes were placed in-between those with negative and positive votes. To create the train, validation and test sets, we split the 4-core dataset according to the user voting data (where each user can vote for one or more product reviews). For each user, we created a list of products they interacted with by considering the reviews they voted for. We then split the products into 60%, 20%, and 20% for the train, validation, and test sets, respectively.

6.3 Metrics

We evaluate review recommendation performance using three common metrics:

- **Normalized Discounted Cumulative Gain (NDCG)**: measures ranking quality by considering both the order and the actual rating value of the reviews.
- **Recall**: measure the proportion of correctly identified reviews with a positive vote in the top-k recommendations.
- **Hit Ratio**: measure whether at least one review with a positive vote appears within the top-k recommendations.

A formal definition of these metrics is provided in Appendix C. We opt to use Hit Ratio in addition to Recall to assess whether we can successfully recommend at least one review the user liked within the top-k. We report the metrics for top-k recommendations and focus on up to 5 reviews, as we believe this is a reasonable number of reviews a user is likely to read on a product page.

7 Results

In this section, we present the results of our proposed approach. Table 6 depicts the results of 12 different models across five categories of the Ciao dataset. The table is segmented into five sections according to model types: non-personalized, content-based, collaborative filtering, deep learning, and graph-based models which includes our suggested *MAGLLM* method. For each category, the models are tested with each of the three signals: *vote*, *write*, and *both.* We report the results using NDCG@5, Recall@1,5, and Hit@1,5 metrics.

Voting vs. Authorship. The results demonstrate that the *vote* signal consistently outperforms the *write* signal in the vast majority of the inspected models (except for Games in the Top Terms method, and DVDs in the NRMS method). In all categories, the gap between the *vote* signal and the *write* signal is substantial, with an improvement of over 20% in the NDCG@5 metric when using collaborative filtering and graph-based methods. This uplift is also observed in Games and Internet categories in the DeepFM method. Similarly, for the Hit@1 metric, there is an improvement of over 20% across all categories and methods, except for the content-based approaches and NRMS. In many cases, combining both signals further boosts performance, particularly in CF methods and DeepFM, but the boost is not very large, up to 6.8% uplift for the DeepFM model in the Beauty and Food & Drink categories. These findings confirm our hypothesis that the voting signal provides an advantage over the authorship signal across a range of recommendation models and categories, and more accurately captures user preferences.

Methods Performance. Inspecting the different types of review recommendation methods, as depicted in the different segments of Table 6, reveals various observations. It can be seen that the CF methods achieve higher performance than the content-based methods, as is commonly observed in other studies for non-extremely sparse scenarios [8]. The popularity baseline, as observed in previous studies [23], is notably strong and often outperforms content-based, CF, and even more advanced models like DeepFM. However, more complex models such as NRMS and graph-based methods, which capture the intricate relationships between all involved entities, are able to surpass the popularity baseline. This highlights the value of methods with richer structures that capture detailed and diverse information for the personalization task. The graph-based methods, including graph only (MAG) and our proposed approach *MAGLLM* which combines LLM, achieve the highest performance across all categories when using the *vote* or *both* signals. Specifically, our method *MAGLLM* outperforms all other methods across all categories. In addition, the graph methods present the largest performance gap between the *write* and *vote* signals. This is likely due to the richer information represented in the graph and the ability of meta-paths to capture complex relationships between nodes. For example, looking at relationship between two users, the meta-path *Voter-Review-Voter* (VRV) can be formed because two users can vote for the same review, whereas *Author-Review-Author* (ARA) cannot be formed, as two users cannot be considered as authors of the same review. In most cases, our method *MAGLLM* successfully ranks at least one review the user liked within the top-5 reviews, as the Hit@5 metric reaches at least 90% across all categories. Appendix D details further analysis, inspecting how the quality of personalization varies with voting patterns.

Table 6: Models performance. The best signal result in each model is boldfaced. The best result in a column is underlined.

Category		DVDs					Beauty					Food & Drink					Internet					Games				
Baseline	Signal	N@5	R@1	R@5	H@1	H@5	N@5	R@1	R@5	H@1	H@5	N@5	R@1	R@5	H@1	H@5	N@5	R@1	R@5	H@1	H@5	N@5	R@1	R@5	H@1	H@5
Non-Personalized																										
Random	N/A	0.397	11.25%	53.77%	16.41%	60.97%	0.447	13.77%	65.53%	18.17%	70.29%	0.411	12.49%	57.91%	17.11%	64.03%	0.336	9.36%	42.60%	14.00%	49.32%	0.413	11.90%	56.95%	17.12%	62.59%
Popularity		0.583	25.84%	72.68%	34.16%	79.31%	0.620	29.37%	81.10%	36.22%	85.12%	0.562	24.89%	72.23%	32.20%	78.02%	0.545	24.78%	63.97%	32.90%	72.05%	0.613	30.46%	75.36%	38.42%	80.35%
Content Based																										
Word2Vec	vote	**0.485**	**17.20%**	**63.37%**	**23.77%**	**70.69%**	**0.532**	**20.67%**	**73.86%**	**26.14%**	**78.54%**	**0.470**	**16.63%**	64.04%	**22.23%**	70.22%	**0.426**	**14.27%**	**52.66%**	**20.54%**	**60.52%**	**0.506**	19.53%	65.68%	26.18%	71.43%
	write	0.467	15.71%	61.53%	21.99%	68.88%	0.513	18.80%	72.38%	24.03%	77.06%	0.457	15.30%	63.01%	20.77%	69.14%	0.416	13.79%	51.67%	19.79%	59.58%	0.491	17.24%	65.55%	23.39%	71.16%
	both	**0.485**	17.09%	63.36%	23.66%	**70.69%**	**0.532**	20.66%	73.75%	26.13%	78.44%	**0.470**	16.60%	**64.08%**	22.21%	**70.28%**	**0.426**	**14.27%**	**52.67%**	**20.54%**	**60.52%**	**0.506**	**19.59%**	**65.70%**	**26.25%**	**71.47%**
Top Terms	vote	**0.447**	**14.47%**	**59.31%**	**20.48%**	**66.68%**	**0.501**	17.90%	70.97%	23.03%	**75.70%**	0.440	14.15%	61.51%	19.21%	67.67%	0.386	11.66%	48.37%	17.14%	56.05%	0.453	15.12%	60.84%	20.99%	66.62%
	write	0.442	14.10%	58.84%	19.99%	66.17%	0.492	16.78%	70.61%	21.67%	75.37%	0.434	13.80%	60.91%	18.77%	67.11%	0.384	11.73%	48.25%	16.96%	55.83%	**0.472**	**15.92%**	**63.36%**	**21.73%**	**68.94%**
	both	**0.447**	14.45%	59.27%	20.47%	66.65%	**0.501**	**17.94%**	**70.98%**	**23.06%**	75.66%	**0.441**	**14.17%**	**61.62%**	**19.25%**	**67.79%**	**0.387**	**11.76%**	**48.46%**	**17.28%**	**56.13%**	0.453	15.17%	60.77%	21.06%	66.52%
Sentence-BERT	vote	**0.469**	**16.44%**	**61.20%**	**22.96%**	**68.56%**	**0.571**	**20.16%**	**75.18%**	**29.21%**	**82.66%**	**0.458**	**15.72%**	**62.80%**	**21.14%**	**69.07%**	**0.406**	12.96%	**50.73%**	18.72%	**58.50%**	0.472	**16.71%**	62.25%	**22.95%**	68.09%
	write	0.460	15.62%	60.54%	21.91%	67.86%	0.502	17.89%	71.30%	22.93%	76.06%	0.449	14.72%	62.12%	20.05%	68.34%	0.399	12.87%	49.92%	18.40%	57.54%	**0.477**	16.27%	**64.04%**	22.15%	**69.64%**
	both	0.468	16.42%	61.17%	22.94%	68.53%	0.514	19.29%	72.01%	24.58%	76.75%	**0.458**	15.63%	62.72%	21.03%	69.00%	**0.406**	**13.03%**	50.70%	**18.80%**	58.45%	0.472	16.69%	62.35%	22.92%	68.19%
KNN Collaborative Filtering																										
KNN item-based	vote	**0.510**	**19.74%**	**65.51%**	**26.53%**	**72.59%**	0.556	23.39%	75.37%	29.28%	79.89%	0.471	16.12%	65.05%	21.69%	70.76%	0.466	18.59%	**56.18%**	25.05%	**63.89%**	0.539	22.93%	68.67%	29.96%	74.14%
	write	0.404	11.43%	54.80%	16.84%	61.91%	0.452	14.41%	65.83%	19.08%	70.61%	0.414	12.43%	58.43%	17.10%	64.42%	0.344	9.37%	43.86%	14.09%	50.81%	0.417	12.70%	56.81%	18.17%	62.41%
	both	0.508	**19.74%**	65.42%	26.13%	72.17%	**0.641**	**32.19%**	**82.18%**	**38.93%**	**86.33%**	**0.559**	**24.24%**	**72.73%**	**30.69%**	**78.27%**	**0.469**	**19.21%**	56.11%	**25.79%**	63.30%	**0.578**	**28.24%**	**70.77%**	**35.55%**	**76.03%**
KNN user-based	vote	**0.530**	**21.84%**	**66.82%**	**28.70%**	**74.02%**	0.589	26.92%	77.54%	32.84%	82.01%	0.527	20.90%	70.08%	26.79%	76.01%	**0.468**	**18.94%**	56.11%	**25.25%**	**63.90%**	0.512	22.49%	63.86%	29.53%	69.56%
	write	0.403	11.39%	54.84%	16.68%	62.00%	0.452	14.09%	66.12%	18.54%	71.08%	0.411	12.24%	58.19%	16.81%	64.42%	0.347	9.59%	44.18%	14.43%	51.18%	0.411	11.93%	56.57%	17.39%	62.10%
	both	0.482	16.76%	63.85%	22.68%	70.94%	**0.606**	**26.96%**	**80.93%**	**33.00%**	**85.26%**	**0.564**	**23.94%**	**73.83%**	**30.36%**	**79.49%**	0.465	18.07%	**56.41%**	24.37%	63.87%	**0.542**	**24.76%**	**67.68%**	**31.07%**	**73.23%**
Deep Learning																										
DeepFM	vote	0.508	18.53%	66.50%	25.24%	73.44%	0.558	22.08%	76.91%	27.90%	81.27%	0.485	17.15%	66.45%	23.04%	72.35%	0.474	17.52%	58.69%	23.95%	66.37%	0.547	22.90%	**70.07%**	**30.09%**	75.37%
	write	0.442	13.90%	58.93%	19.71%	66.14%	0.473	16.26%	67.65%	21.26%	72.46%	0.431	13.80%	60.02%	18.91%	66.07%	0.341	9.24%	43.21%	14.18%	50.20%	0.419	13.17%	56.73%	18.81%	62.29%
	both	**0.525**	**19.78%**	**68.11%**	**26.64%**	**75.16%**	**0.596**	**26.26%**	**79.56%**	**32.65%**	**83.88%**	**0.519**	**20.03%**	**69.37%**	**26.28%**	**75.22%**	**0.496**	**21.03%**	**58.71%**	**28.10%**	**66.62%**	**0.554**	**24.50%**	69.80%	20.06%	**75.38%**
NRMS	vote	0.681	36.35%	80.59%	46.06%	86.66%	**0.731**	**42.54%**	**88.71%**	**50.68%**	**92.21%**	**0.696**	**37.56%**	**84.56%**	**46.58%**	**89.38%**	0.604	29.85%	**69.35%**	39.15%	**77.59%**	**0.647**	**33.23%**	**77.94%**	**41.70%**	**83.48%**
	write	0.692	37.67%	**81.34%**	47.64%	**87.27%**	0.724	41.31%	88.50%	49.51%	91.90%	0.690	36.59%	84.28%	45.82%	88.99%	0.596	29.36%	68.46%	38.73%	76.82%	0.598	27.74%	74.38%	35.49%	79.91%
	both	**0.693**	**37.83%**	81.23%	**47.74%**	87.20%	0.725	41.54%	88.43%	49.76%	91.90%	0.694	37.35%	84.44%	46.52%	89.17%	**0.606**	**30.47%**	68.89%	**39.83%**	77.33%	0.615	30.49%	75.16%	38.51%	80.64%
Graphs																										
MAG	vote	**0.876**	**70.42%**	**88.80%**	**84.89%**	**90.59%**	**0.898**	**74.74%**	**92.58%**	**86.22%**	**93.54%**	0.837	67.93%	85.98%	80.78%	87.08%	**0.881**	**69.74%**	**87.14%**	**85.89%**	**90.04%**	0.882	71.65%	**89.94%**	83.54%	**92.24%**
	write	0.704	45.45%	80.38%	54.56%	84.89%	0.689	43.21%	84.07%	49.93%	87.11%	0.664	41.11%	80.01%	48.32%	83.67%	0.604	34.72%	68.97%	42.55%	75.20%	0.723	52.13%	78.65%	61.34%	82.05%
	both	0.856	67.67%	87.22%	81.89%	89.18%	0.865	71.47%	89.84%	82.76%	90.77%	**0.843**	**68.34%**	**86.70%**	**81.25%**	**87.86%**	0.862	67.81%	85.67%	83.66%	88.34%	**0.883**	**72.00%**	89.69%	**84.65%**	91.81%
MAGLLM	vote	**0.883**	**71.29%**	**89.29%**	**85.39%**	**91.15%**	**0.911**	**76.96%**	**93.29%**	**88.38%**	**94.23%**	0.858	69.41%	88.46%	82.13%	89.75%	**0.894**	**70.65%**	**89.46%**	**86.64%**	**92.30%**	**0.912**	**75.48%**	**92.07%**	**87.91%**	**94.29%**
	write	0.669	42.60%	77.12%	51.73%	81.07%	0.680	41.04%	84.17%	47.40%	87.30%	0.639	40.01%	76.91%	47.44%	79.95%	0.616	39.59%	67.72%	50.00%	71.70%	0.745	54.17%	79.84%	64.53%	83.35%
	both	0.850	68.44%	86.20%	82.26%	87.94%	0.875	72.08%	90.93%	83.20%	91.93%	**0.863**	**70.40%**	**88.69%**	**83.17%**	**89.91%**	0.857	65.75%	86.70%	81.23%	89.88%	0.899	74.87%	90.26%	87.20%	92.39%

8 Discussion and Future Work

We present, for the first time in review personalization, the use of voting signals to learn user preferences. We compare the use of review authorship and review voting signals for personalization across popular recommendation methods and five different e-commerce domains. Our results indicate that utilizing the voting signal consistently achieves substantially higher personalization performance compared to the authorship signal. In some cases, combining authorship and voting signals results in additional improvements compared to using the voting signal alone. We also suggested *MAGLLM*, a personalized review recommendation technique that models relationships among users, products, and reviews, using a heterogeneous graph network with meta-paths and improves user representation using LLM. *MAGLLM* reaches high results, with Hit@5 varying from 89.9% (Food & Drink) to 94.2% (Beauty and Games), which allows high quality personalization.

Our results highlight the advantage of using voting data over authored data, as its availability and the strength of its signal contribute to more effective personalization. These findings confirm our hypothesis that the voting signal can more accurately reflect user preferences. Our analysis shows that on e-commerce platforms, the number of votes is up to double the number of reviews. This suggests that there is a significant potential for growth in user engagement through voting, as the current volume of votes is relatively low, although still much higher than authored reviews. Encouraging user feedback is a key for unlocking this potential. While writing a review is often time-consuming and requires more effort, providing direct feedback through voting mechanisms, such as likes, dislikes, or numeric ratings, is much simpler and more intuitive for many users. This ease of engagement allows for a broader spectrum of user participation.

The performance improvement achieved by using the voting signal highlights its significant potential for enhancing personalized recommendations. Based on these findings, we propose key practical implications for e-commerce platform design. First, make the voting-for-review feature visible and prominent to encourage more voting. Some platforms may not invest enough in this feature or lack it altogether, but it can be a powerful tool to improve recommendation systems in various contexts. Second, consider introducing more detailed voting options, such as star ratings or multi-dimensional feedback similar to Yelp's categories (e.g., "helpful," "thanks," "love this," and "oh no"), to better capture nuanced user preferences and enhance personalization. Third, emphasize aggregate vote counts per review: highlighting how many users found a review helpful can encourage engagement and improve personalization data reliability. Finally, as generative AI becomes more prevalent, automated review generation is likely to become more popular. While this may reduce the need for users to write full reviews, voting will remain a valuable form of explicit feedback. Therefore, investing in this feature could be highly beneficial.

For future work, several directions can extend the paper. Exploring other graph-based approaches could further uncover the potential of heterogeneous graphs in review recommendation. Analyzing meta-path importance could optimize graph representation by identifying key relationships. Another direction is to investigate whether voting, rather than authorship, improves cold-start recommendations. Leveraging votes could reduce the "warming" time and better identify similar users during the cold-start phase. Finally, in-vivo experiments within live environments could offer valuable insights into the real-world applicability of our approach. To measure the success of the approach, user engagement can be evaluated at both the review and the platform level. Review level metrics such as reading time, voting interactions, and scrolling depth can indicate user interest and satisfaction with recommendations. At the platform level, metrics like session duration, conversion rates, changes in review writing and voting activity over time, and overall user activity could reflect improved user experiences.

References

[1] 2024. MAGNN. https://github.com/cynricfu/MAGNN/tree/master

[2] 2024. Python bindings. https://github.com/abetlen/llama-cpp-python

[3] 2024. A Python scikit for building and analyzing recommender systems. https://github.com/NicolasHug/Surprise

[4] 2024. Pytorch Implementation of EMNLP 2019 NRMS. https://github.com/aqweteddy/NRMS-Pytorch

[5] 2024. PyTorch package of deep-learning based CTR models. https://github.com/shenweichen/DeepCTR-Torch

[6] 2024. Sentires: A Toolkit for Phrase-level Sentiment Analysis. https://github.com/evison/Sentires

[7] Deepak Agarwal, Bee-Chung Chen, and Bo Pang. 2011. Personalized recommendation of user comments via factor models. In *Proceedings of the 2011 conference on empirical methods in natural language processing*. 571–582.

[8] Parul Aggarwal, Vishal Tomar, and Aditya Kathuria. [n. d.]. Comparing content based and collaborative filtering in recommender systems. *International Journal of New Technology and Research* 3, 4 ([n. d.]), 263309.

[9] Wenshuo Chao, Zhi Zheng, Hengshu Zhu, and Hao Liu. 2024. Make Large Language Model a Better Ranker. *arXiv:2403.19181* (2024).

[10] Fermín L Cruz, José A Troyano, Fernando Enríquez, F Javier Ortega, and Carlos G Vallejo. 2013. 'Long autonomy or long delay?'The importance of domain in opinion mining. *Expert Systems with Applications* 40, 8 (2013), 3174–3184.

[11] Anupam Dash, Dongsong Zhang, and Lina Zhou. 2021. Personalized ranking of online reviews based on consumer preferences in product features. *International Journal of Electronic Commerce* 25, 1 (2021), 29–50.

[12] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805* (2018).

[13] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*. 2331–2341.

[14] Kavita Ganesan and ChengXiang Zhai. 2012. Opinion-based entity ranking. *Information retrieval* 15, 2 (2012), 116–150.

[15] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv:1703.04247* (2017).

[16] Lei Guo, Enhua Tan, Songqing Chen, Xiaodong Zhang, and Yihong (Eric) Zhao. 2009. Analyzing patterns of user content generation in online social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. 369–378.

[17] Ido Guy. 2022. Social Recommender Systems. *Recommender Systems Handbook* (2022), 835–870.

[18] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 230–237.

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

[20] Chunli Huang, Wenjun Jiang, Jie Wu, and Guojun Wang. 2020. Personalized review recommendation based on users' aspect sentiment. *ACM Transactions on Internet Technology (TOIT)* 20, 4 (2020), 1–26.

[21] Reda Igebaria, Eran Fainman, Sarai Mizrachi, Moran Beladev, and Fengjun Wang. 2024. Enhancing Travel Decision-Making: A Contrastive Learning Approach for Personalized Review Rankings in Accommodations. *arXiv:2407.00787* (2024).

[22] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2020. A Re-visit of the Popularity Baseline in Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 1749–1752.

[23] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2020. A Re-visit of the Popularity Baseline in Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1749–1752. https://doi.org/10.1145/3397271.3401233

[24] Farhad Khalilzadeh and Ilyas Cicekli. 2024. REHREC: Review Effected Heterogeneous Information Network Recommendation System. *IEEE Access* 12 (2024), 42751–42760.

[25] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[26] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. 939–948.

[27] Huiting Liu, Yi Chen, Peipei Li, Peng Zhao, and Xindong Wu. 2023. Enhancing review-based user representation on learned social graph for recommendation. *Knowledge-Based Systems* 266 (2023), 110438.

[28] Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint* cs/0205028 (2002).

[29] Tianhui Ma, Yuan Cheng, Hengshu Zhu, and Hui Xiong. 2023. Large Language Models are Not Stable Recommender Systems. *arXiv preprint arXiv:2312.15746* (2023).

[30] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 142–150.

[31] Muhammad Faraz Manzoor, Adnan Abid, Naeem A Nawaz, and Atif Alvi. 2022. Aspect based sentence segregated dataset of hybrid car's consumers online reviews. *Data in Brief* 42 (2022), 108293.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781* (2013).

[33] Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. 2011. Review recommendation: personalized prediction of the quality of online reviews. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. 2249–2252.

[34] Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. 2012. Etf: extended tensor factorization model for personalizing prediction of review helpfulness. In *Proceedings of the fifth ACM international conference on Web search and data mining*. 163–172.

[35] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*. 188–197.

[36] Akhil Sai Peddireddy. 2020. Personalized Review Ranking for Improving Shopper's Decision Making: A Term Frequency based Approach. *arXiv:2009.03258* (2020).

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

[38] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. 2022. A review-aware graph contrastive learning framework for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1283–1293.

[39] Vaishak Suresh, Syeda Roohi, Magdalini Eirinaki, and Iraklis Varlamis. 2014. Using Social Data for Personalizing Review Rankings.. In *RSWeb@ RecSys*.

[40] Lei Tan, Daofu Gong, Jinmao Xu, Zhenyu Li, and Fenlin Liu. 2023. Meta-path fusion based neural recommendation in heterogeneous information networks. *Neurocomputing* 529 (2023), 236–248.

[41] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: Discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*. 93–102.

[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv:2302.13971* (2023).

[43] Bingkun Wang, Yulin Min, Yongfeng Huang, Xing Li, and Fangzhao Wu. 2013. Review rating prediction based on the content and weighting strong social relation of reviewers. In *Proceedings of the 2013 international workshop on Mining unstructured big data using natural language processing*. 23–30.

[44] Haiming Wang, Wei Liu, and Jian Yin. 2021. Multi-Task Learning with Personalized Transformer for Review Recommendation. In *WISE*. Springer, 162–176.

[45] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*. 501–508.

[46] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091* (2023).

[47] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *EMNLP-IJCNLP*. 6389–6394.

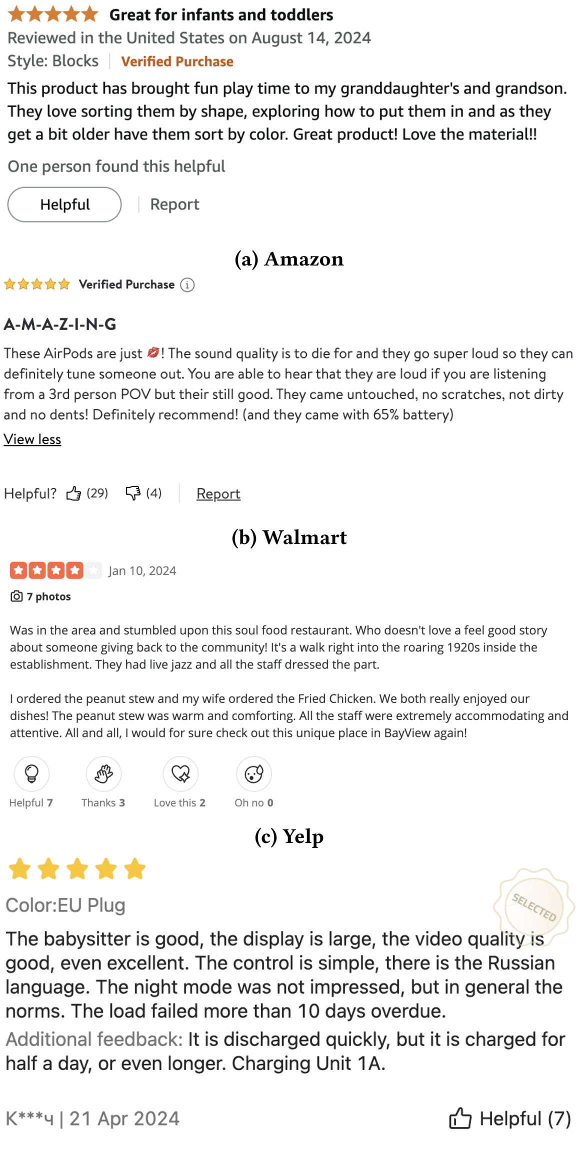
[48] Yelp. 2024. Yelp Dataset. https://www.yelp.com/dataset

[49] Yijiong Yu, Huiqiang Jiang, Xufang Luo, Qianhui Wu, Chin-Yew Lin, Dongsheng Li, Yuqing Yang, Yongfeng Huang, and Lili Qiu. 2024. Mitigate Position Bias in Large Language Models via Scaling a Single Dimension. *arXiv:2406.02536* (2024).

[50] Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023. LlamaRec: Two-stage recommendation using large language models for ranking. *arXiv:2311.02089* (2023).

A Screenshots of Review Feedback Functionality on various e-commerce Platforms.

Figure 3 demonstrates the review feedback functionality across four popular e-commerce platforms. There are subtle differences between these platforms. For instance, review feedback can be a simple "helpful" indicator (e.g., Amazon, Aliexpress), a thumbs up or thumbs down for "liking" or "disliking" (Walmart), or multi-dimensional with "helpful", "thanks", "love this", and "oh no" (Yelp).



★★★★★ **Great for infants and toddlers**
Reviewed in the United States on August 14, 2024
Style: Blocks | Verified Purchase

This product has brought fun play time to my granddaughter's and grandson. They love sorting them by shape, exploring how to put them in and as they get a bit older have them sort by color. Great product! Love the material!!

One person found this helpful

Helpful | Report

(a) Amazon

★★★★★ **Verified Purchase** ⓘ

A-M-A-Z-I-N-G

These AirPods are just 💋! The sound quality is to die for and they go super loud so they can definitely tune someone out. You are able to hear that they are loud if you are listening from a 3rd person POV but their still good. They came untouched, no scratches, not dirty and no dents! Definitely recommend! (and they came with 65% battery)

View less

Helpful? 👍 (29) 👎 (4) | Report

(b) Walmart

★★★★☆ Jan 10, 2024
📷 7 photos

Was in the area and stumbled upon this soul food restaurant. Who doesn't love a feel good story about someone giving back to the community! It's a walk right into the roaring 1920s inside the establishment. They had live jazz and all the staff dressed the part.

I ordered the peanut stew and my wife ordered the Fried Chicken. We both really enjoyed our dishes! The peanut stew was warm and comforting. All the staff were extremely accommodating and attentive. All and all, I would for sure check out this unique place in BayView again!

Helpful 7 Thanks 3 Love this 2 Oh no 0

(c) Yelp

★★★★★
Color:EU Plug

The babysitter is good, the display is large, the video quality is good, even excellent. The control is simple, there is the Russian language. The night mode was not impressed, but in general the norms. The load failed more than 10 days overdue.
Additional feedback: It is discharged quickly, but it is charged for half a day, or even longer. Charging Unit 1A.

K***ч | 21 Apr 2024 👍 Helpful (7)

(d) Aliexpress

Figure 3: Review feedback examples on different e-commerce platforms

B LLM Experiments for Our Suggested Approach

B.1 LLM Prompts for User Profile

Figure 4 illustrates some of the prompts we used in the experimentation of generating user profile using LLM based on the user history, which is the *signal-based review set* associated with the user.

```
system: You are required to generate user profile based on the
history of a user. The profile should contain only user
interests that can be learned from the given history. Do not
infer the user name, age or gender.
The profile will later be used to calculate personalized
recommendation of new reviews. Use up to 300 tokens.
prompt: The user previously {signal} the following reviews:
<history>.
His profile:
```

```
system: Your objective is to create user profile using their
review history. The profile should be general, without any
personal details, but with enough details to allow
personalized recommendations of new reviews.
The profile should contain up to 300 tokens.
prompt: The user previously {signal} the following reviews:
<history>.
His profile:
```

Figure 4: LLM prompt examples for generating user profile using the user history

B.2 LLM Prompt Tuning

We explore various alternatives for setting prompt parameters to generate user profiles, focusing on parameters such as review text size, history size, review order, zero-shot and few-shot prompting, and Chain-of-Thought (CoT) reasoning. Experimentation results, depicted in detail in Table 7, indicate that the best performance is achieved with a review text size of 150 tokens and a history size of 5 reviews. Including more than 5 reviews does not lead to further performance improvements. For the order of reviews in the user history, sorting them by quality rating in ascending order, with the highest-rated reviews placed last, produces the best outcome. This finding suggests the presence of a position bias as has been observed in other works [9, 19]. We also experiment with zero-shot and few-shot prompting using one and two examples. The examples include two variations: (1) a user's review history with a natural language user profile based on these reviews, and (2) only the user profile without the history. Few-shot prompting with two examples including only user profile achieves the highest performance. In CoT reasoning, we test two strategies of zero-shot: the first involves adding "*Let's think step by step*" to the prompt [25], while the second builds on this by including an additional prompt extension, "*Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.*" [46]. However, incorporating reasoning results in lower performance compared to not using reasoning at all. This could be due to reasoning prompts introducing unnecessary complexity or deviation from the concise nature required for user profile generation.

Table 7: Prompt Tuning.

Parameter	Variants	Best
Review text size	100, 150, 200	150 tokens
History size	5, 7, 10, 12, 15	5 reviews
Order of reviews in user history	Random, Highest score first, Highest score last	Highest score last
Zero-shot vs. few-shot prompting	Zero-shot, Few-shot with one example, Few-shot with two examples	Few-shot with two examples
Chain-of-Thought reasoning	No reasoning, Zero-shot Chain-of-Thought [25], Plan-and-Solve prompting [46]	No reasoning

B.3 Exploring a Two-Stage LLM-Based Recommendation Approach

Other than for user profile generation, we explored an alternative strategy for leveraging LLMs to our task by applying a two-stage approach consisting of retrieval and recommendation stages. This concept, introduced in the work of LlamaRec, was applied for sequential recommendation [50]. In the retrieval stage, a model is used to generate an initial list of candidates, and in the recommendation stage, an LLM ranks these candidates. Similarly, we used a recommendation model, specifically the heterogeneous graph-based model with meta-paths used in our approach to identify the initial candidates, and then the LLM re-ranked the retrieved candidates. Initially, we applied a list-wise approach, but since LLMs have been shown to suffer from position bias [9, 19, 29, 49], where the model gives disproportionate importance to items based on their position in the input sequence, we also experimented with a point-wise approach, in which the LLM ranked one candidate at a time. However, even with the point-wise approach, the performance remained low, actually degrading the performance of the recommendation model used at the retrieval stage. We therefore do not report the results of this approach in detail.

C Metrics

We use three common metrics to evaluate review recommendation performance:

- **Normalized Discounted Cumulative Gain (NDCG)**: measures ranking quality in recommendations systems. It takes into account the rank position information and the actual value of the rating. NDCG is calculated as:

$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{1}$$

$$DCG@K = \sum_{i=1}^{k} \frac{qrat_i}{log_2(i+1)} \tag{2}$$

  where $qrat_i$ is the quality rating score of the review at position i IDCG refer to ideal DCG representing the ideal order of ranking (e.g., reviews that the user votes with 5 at the top, followed by 4, and so on until reviews with a vote of 0, where reviews not voted for by the user are in-between 2 and 3).

- **Recall**: measures the proportion of correctly identified reviews with a positive vote in the top-k recommendations out of the total number of reviews with a positive vote. It is calculated as follows:

$$Recall@K = \frac{\text{Number of reviews with a positive vote in top-k}}{\text{Total number of reviews with a positive vote}} \tag{3}$$

- **Hit Ratio**: measures whether at least one positive review appears within the top-k recommended reviews. It is calculated as follows:

$$Hit@K = \frac{\text{Number of reviews with a positive vote in top-k}}{\text{Total number of reviews}} \tag{4}$$

D Voting Behavior Analysis

We explore how the quality of personalization varies with voting patterns. This analysis is conducted using the best-performing model, *MAGLLM*, with the *vote* signal, focusing only on Beauty category. First, we examine the personalization performance according to the user's *primary vote*, i.e., the most common score across all of the votes performed by the user. Table 8 presents the results across the five metrics, for primary votes of 3, 4, and 5. It can be seen that all metrics are higher when the primary vote is higher. This indicates that the voting signal for personalization is stronger for users who tend to vote more positively. In other words, votes with a higher score reflect a stronger preference from the user to the review, as could be intuitively expected, and thus is a stronger indication for the model. We also examine personalization performance as a factor of the user's total number of votes (regardless of their score). We found that performance was quite similar across users with different number of total votes, with only a slight and inconsistent increase for users who are more actively voting. For example, NDCG@5 increases from 0.875 for users with 1 total vote to 0.909 for users with 10 or more votes. Recall@1 even slightly decreased, from 81.08% for users with 1 vote to 80.25% for users with 10 votes or more. These results indicate that our personalization is effective even for users with very few votes. Finally, we examine the impact of users voting for a single review versus multiple reviews within a product. The model performs slightly better when there are multiple votes for a product's reviews. However, even with just one vote, the personalization remains effective, successfully ranking the review the user preferred at the top. The NDCG@5 scores are 0.907 for a single review and 0.925 for multiple reviews, while the Hit@1 scores are 87.96% and 89.67%, respectively.

Table 8: Performance of *MAGLLM* by the user's primary vote.

Primary Vote	N@5	R@1	R@5	H@1	H@5
3	0.813	74.65%	84.73%	77.65%	85.42%
4	0.899	83.07%	92.35%	87.10%	92.82%
5	0.914	86.51%	96.18%	86.78%	96.18%