Lookalike Audience Expansion: A Graph-Based Model with LLMs

Rumana Ferdous Munne[1], Md Mostafizur Rahman[2] and Yuji Matsumoto[1]

[1]*RIKEN-AIP, Tokyo, Japan.*
[2]*Rakuten Group, Inc., Tokyo, Japan.*

Abstract

Lookalike modeling is the key to digital marketing, driving product sales and improving ad campaigns by identifying users similar to a given set of seed users. However, this task presents several challenges. Companies often handle hundreds of marketing campaigns daily, targeting a large user base, making it difficult for models that depend solely on high-level features to achieve optimal performance. Additionally, the limited size of seed lists can lead to over-fitting, requiring models to generalize effectively. Traditional methods, using deep learning and graph-based approaches, excel at capturing complex user-item relationships but heavily depend on ID-based data and often overlook valuable textual information, such as user reviews and item descriptions. Moreover, privacy concerns and increasing data regulations further complicate the process, as conventional models frequently rely on sensitive user attributes. To overcome these challenges, we propose a Graph-Lookalike Model (GLoM) that integrates large language models (LLMs) into lookalike modeling. GLoM enhances user targeting by combining advanced representation learning with LLMs, capturing important semantic information in user behavior, sentiments, and preferences, while preserving the graph structure and incorporating auxiliary textual features. Our experiments show that GLoM successfully expands the user base across diverse categories like books, movies, electronics, and automotive, outperforming the baselines.

Keywords

Lookalike modeling, User targeting, Embedding learning, Representation learning, LLMs

1. Introduction

The rapid expansion of the internet has led to a significant increase in digital marketing activities, with a large number of users interacting with these activities on a daily basis. In this vast online marketplace with billions of users, it is crucial for marketers to deliver content, ads, or products to the right audience through recommendation systems or advertising platforms. Lookalike modeling plays a key role in identifying similar users to a given set of seed users (Figure 1), thus increasing the chances of achieving specific marketing goals. Leading tech companies like Facebook, Google [1], Tencent [2], and LinkedIn [3] have developed robust platforms for such campaigns, yet the task remains complex.

Lookalike model offers significant economic benefits by identifying high-potential users for marketing campaigns. Traditional methods, which rely on demographic data or purchasing behavior, often miss latent users and depend solely on implicit feedback. The lack of generalization,

rumanaferdous.munne@riken.jp (R. F. Munne); mdmostafizu.a.rahman@rakuten.com (M. M. Rahman); yuji.matsumoto@riken.jp (Y. Matsumoto)

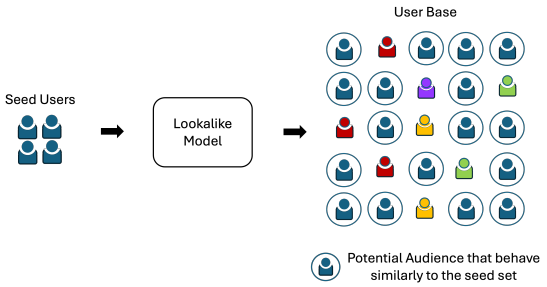

Figure 1: Overview of Lookalike Audience Expansion.

user sentiment understanding, and insufficient seed users make things more challenging. Scaling these models to meet campaign needs is another significant challenge. An effective model must capture both explicit and implicit user traits, incorporate user sentiments, and scale efficiently. Deep learning [4, 5] and graph-based algorithms [6, 7], commonly used in recommender systems, have shown promise for lookalike modeling. Graph-based models [6, 7] have demonstrated remarkable capabilities in capturing complex user-item relationships. However, these models often operate on mapped user/item information for learning. The use of demographic or location-based data also raises privacy concerns. Moreover, the primary reliance on mapped data in graph-based models may overlook valuable information, such as the rich textual content associated with users and items.

An ideal lookalike model should go beyond basic user preferences and incorporate hidden factors such as user sentiments and user experiences. Large Language Models (LLMs) can play a pivotal role in analyzing user reviews and ratings to identify patterns of user-user similarity based on text data. Understanding how users feel about their purchases provides deeper insights while reducing reliance on private information. The proposed GLoM model addresses these limitations by avoiding the use of private user data and instead utilizing purchase interactions alongside publicly available data, such as product reviews and LLM-generated user or item profiles. An effective lookalike model should account not only for users preferences but also for factors such as affordability and lifestyle. For instance, a user purchasing electric car accessories might also be interested in smart home devices, such as Amazon Alexa, smart thermostats, or energy management hubs. This suggests that while the user actively engages with automotive products, they may have relevant interests in other domains where they lack a purchase history. It also highlights the potential for targeting users who can afford premium items but remain overlooked if the model focuses only on single-domain data. To address these limitations, our approach incorporates users' complex cross-domain behaviors and item similarity patterns to deliver more comprehensive and effective audience expansion for user targeting.

Understanding complex relationships between users and items, as well as identifying users with similar behaviors, requires effective data structures like Knowledge Graphs (KGs). KGs represent information as triples, where each triple encodes a factual relationship. The proposed GLoM model utilizes a knowledge graph constructed from user-item interactions, generated user and item polarity (i.e., whether a user likes or dislikes an item), and user-user similarity connections. LLMs excel at capturing the semantic meaning of entities, relationships, and text-encoded triples.

However, they struggle to model the structural relationships inherent in graph data. Conversely, Graph Neural Networks (GNNs) are well-suited for processing graph structures but lack the ability to fully grasp the rich textual semantics that LLMs handle effectively. By combining the strengths of both LLMs and GNNs, GLoM creates a comprehensive model that integrates structural and semantic insights, enabling accurate user behavior analysis and effective lookalike modeling.

GLoM integrates a pre-trained embedding model with a graph convolutional network to identify lookalike users. During pre-training, user and item profiles generated by LLMs are leveraged with a knowledge graph. This pre-training and fine-tuning paradigm allows GLoM to extract informative and transferable knowledge from abundant unlabelled data through self-supervision tasks such as masked language modeling. This approach is particularly beneficial when the labeled seed list for user targeting is insufficient, as it avoids the need to train a new model from scratch, maintaining the integrity of the pre-trained model. GLoM employs three different aggregation techniques (please refer to Sec. 3.8.1) for node features, enhancing the pre-trained model with effective smoothing. This approach mitigates issues such as oversmoothing and irrelevant smoothing, thereby improving the precision of lookalike modeling. The main contributions of our work are organized as follows:

- We propose a novel two-stage model, GLoM, which leverages LLMs and KGs for the lookalike audience expansion problem to demonstrate its effectiveness and robustness.
- To the best of our knowledge, GLoM is the first lookalike model that combines the strengths of Large Language Models (LLMs) for semantic understanding and Graph Neural Networks (GNNs) for structural analysis.
- We introduce a pre-training and fine-tuning paradigm that utilizes self-supervised tasks, to extract transferable knowledge from abundant unlabeled data, reducing dependency on labeled seed lists.
- GLoM employs Large Language Models (LLMs) to generate user and item profiles, along with user-item polarity (e.g., whether a user likes or dislikes an item), providing valuable semantic insights for improved lookalike modeling.
- GLoM significantly outperforms state-of-the-art (SOTA) lookalike models across four public datasets.

2. Related Works

In this section, we review related work on lookalike modeling, focusing on various approaches including similarity-based methods, clustering techniques, rule-based methods, multi-task learning, and graph-based models.

Similarity-based methods, such as those proposed by [8], expand a given seed list by calculating the similarity between pairs of seed users and candidate users using predefined metrics like Cosine or Jaccard similarity. Several studies have explored approaches to lookalike modeling, such as k-means clustering [9, 10], which offers simplicity but faces challenges in capturing complex

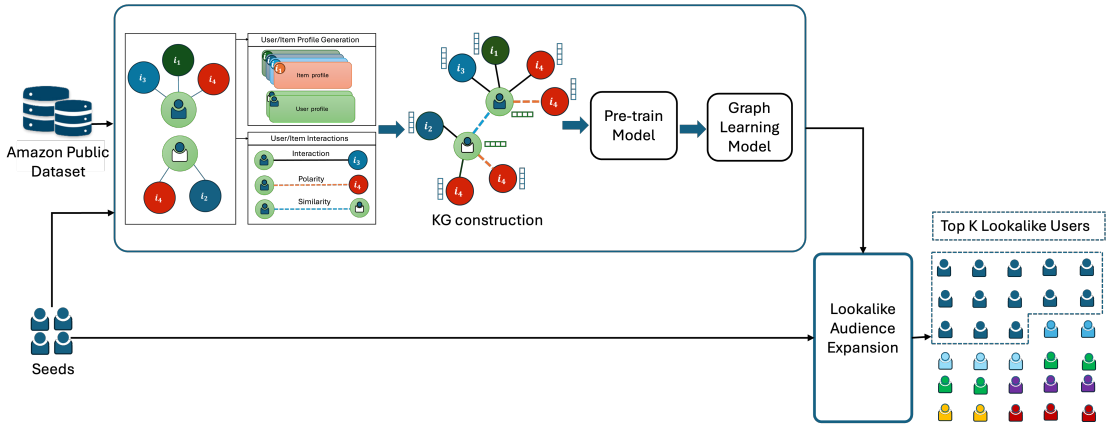

Figure 2: The model structure of the Graph Lookalike Model (GLoM).

and high-dimensional relationships. Clustering-based models are also quite popular when the number of tasks or campaigns is limited [9, 11, 12]. These models primarily cluster users to generate a candidate set, which is then filtered using a regression model. However, this approach compromises on precision and algorithmic complexity to prioritize online performance.

On the other hand, rule-based methods identify similar users based on specific demographic features or interests, as targeted by marketers. These methods typically rely on user profile mining to infer interest tags from user behavior [13]. The limitations of similarity-based and rule-based models are: the former depends on the choice of the similarity function, and the latter captures only high-level features, often leading to suboptimal performance. GLoM addresses these issues by incorporating semantic understanding and more complex relationship modeling.

Multi-task learning has also been explored for lookalike modeling. This approach allows for simultaneous learning across multiple tasks, potentially improving efficiency [14]. However, existing multi-task methods are generally designed for scenarios with fewer than five tasks [11], limiting their applicability in real-world settings where hundreds of marketing campaigns run daily. Model-based methods train customized prediction models for each campaign or task, and GLoM falls into this category. For example, logistic regression (LR) has been utilized to expand audiences, which proved effective [15]. One-stage methods that train models from scratch for each campaign are time-consuming and prone to overfitting. More recently, two-stage approaches [16, 2] have been proposed to pretrain embeddings using data from all campaigns. Rakuten employed a lookalike model for its advertising platform, relying heavily on demographics,user/item attributes and user-item interactions [17]. While it achieves strong performance, comparison is challenging as the data is not publicly available. However, these methods often overlook generalization, task relationships, and semantic understanding. In contrast, GLoM efficiently addresses these challenges.

3. Graph Lookalike Model (GLoM)

3.1. Preliminaries

Problem Statement: In a lookalike setting, a list of m seed users $S_u = (u_1, u_2, \ldots, u_m)$ is given to the model and the task is to find k similar users to the seed list S_u where $k \gg m$.

Knowledge Graph: A Knowledge Graph (KG) is a directed, labeled graph $G = (V, E, R)$ where: V is the set of entities (nodes), R is the set of relations (edge types), and $E \subseteq V \times R \times V$ is the set of triples (v_1, r, v_2) representing directed edges, where $v_1, v_2 \in V$ and $r \in R$.

Our proposed GLoM operates in two stages: a pre-training stage and a graph learning stage. In the pre-training stage, we use LLM-based user/item profile generation and knowledge graph to generate the pre-trained embeddings for the graph learning stage. We construct a knowledge graph using data from user-item interactions, user-item polarity edges (please refer to Sec. 3.3) and user-user similarity edges (please refer to Sec. 3.4). We represent a triplet as (u, r, v). Hereafter, bold lowercase letters indicate embeddings, and bold uppercase letters denote matrices. Figure 2 demonstrates GLoM's model architecture.

3.2. User and Item Profile Generation

In this section, we describe how we create textual descriptions, or profiles, for users and items for GLoM. These profiles improve the understanding of user and item interaction preferences by adding textual information related to them. For user and item profile generation, we use two types of information: one is the input prompt for users or items M_u and M_v, and the other is the general prompt guideline S_u and S_v. So, the user and item profiles are generated as: $P_u = LLM(M_u, S_u), \quad P_v = LLM(M_v, S_v)$. We discuss the process in detail in the sections below.

3.2.1. Input Prompt for User

In the context of user profile generation, Large Language Models can be utilized to effectively encapsulate the particular types of items that users are likely to purchase. By leveraging collaborative filtering, the system generates a user profile P_u by first identifying items I_u with which the user u has interacted. A subset $\hat{I}_u$ of these items is then uniformly sampled. For each item v in this subset, a textual representation $c_v = [\tau, P_v, r_v^u]$ is created, where τ is the title, P_v is the previously generated item profile, and r_v^u is the review provided by the user u. The input prompt M_u for generating the user profile is then defined as $M_u = f_u(\{c_v \mid v \in \hat{I}_u\})$, where $f_u(\cdot)$ organizes these textual attributes into a coherent string. This approach provides a comprehensive representation of the users personalized tastes and preferences, ensuring that the generated profile accurately reflects their real opinions and interests (Figure 3).

3.2.2. Input Prompt for Item

For item profile generation, LLMs can be guided to produce profiles that accurately reflect the appealing characteristics of items. The textual information of an item $v \in V$ is categorized into four types: title τ , original description δ , item attributes $\gamma = \{\gamma_1, \ldots, \gamma_{|\gamma|}\}$, and a collection of n

Input Prompt for User (M_u)

Purchased Items [
Title: Wall-E (Mandarin Chinese Edition)
Category: ['Movies & TV', 'Characters & Series', 'Wall-E']
Description: ["Pixar genius reigns in this funny romantic comedy, which stars a robot who says absolutely nothing for a full 25 minutes yet somehow completely transfixes and endears himself to the audience within th..."]
Review: "Happy to have found this video. How's of children entertainment."

Title: Spy: Susan Cooper Undercover
Category: ['Movies & TV', 'Blu-ray', 'Movies']
Description: ["Quick Shipping !!! New And Sealed !!! This Disc WILL NOT play on standard US DVD player. A multi-region PAL/NTSC DVD player is request to view it in USA/Canada. Please Review Description..."]
Review: "Glad to have found this dvd, funny show."

Title: Rocky Horror Picture Show VHS
Category: ["Movies & TV', 'Art House & International', 'By Country', 'United Kingdom', 'Music & Musicals']
Description: ["Rocky Horror Picture Show [VHS]..."]
Review: "Love this show. Glad I found it on dvd."

Title: Megamind
Category: ['Movies & TV', 'Science Fiction & Fantasy', 'Science Fiction', 'Animation']
Description: [" A smart, funny and original treat, Michael O'Sullivan, The Washington Post, DreamWorks Animations Megamind puts a whole new hilarious twist on the superhero movie...."]
Review: ["Really enjoyed!]

General Prompt Guideline for User (S_u)

Prompt : Generate a user profile based on the items purchased and the reviews provided by the user within 100 words

Guideline:

- Identify which genre of item the user interested in
- Identify what key features he looks in the item
- Note any comparisons, notable features, or unique aspects mentioned

Generated User Profile

GPT 4 Turbo: Based on the reviewed movies, the user enjoys entertaining and humorous films, particularly those suitable for children and family viewing. He appreciate well-crafted animated features and unique or cult classics. The user values quick shipping and format compatibility when purchasing DVDs, and the seek engaging, high-quality content across various genres.

LLaMa3-70B : The user's reviews suggest a strong interest in children's entertainment, prioritizing humor and enjoyment for kids. They appreciate films with unique twists, such as the silent robot in Wall-E. Convenience is also important, with a preference for quick shipping and new, sealed DVDs. Region and format seem non-essential, as they've purchased international DVDs. Their preferred genre is likely science fiction, animation, and comedy. With a focus on family-friendly content, they seek movies that are fun and engaging.

Figure 3: User profile generation.

user reviews $r = \{r_1, \ldots, r_n\}$. The input prompt M_v for generating the item profile is structured as $M_v = f_v(x)$ with respect to $x = [\tau, \delta, \gamma, \hat{r} \subseteq r]$. The function $f_v(\cdot)$ combines these various text features into a single string, ensuring the inclusion of item descriptions or selected user reviews. This ensures that the LLM generates item profiles that accurately capture the distinct attributes and qualities that make the item appealing to users (Figure 4).

3.2.3. User/Item Profile Generation Example

This section presents examples of generating user and item profiles using large language models, with a focus on the Amazon-Movies and TV dataset. While we showcase specific examples from this dataset, the same approach is applied to other datasets such as Books, Electronics, and Automotive, with slight variations in the prompts for general prompt guideline tailored to the item type. For instance, for books, the prompt might ask, "What kind of story would the user

Input Prompt for Item (M_v)

Item data
Title: Megamind
Category: ['Movies & TV', 'Science Fiction & Fantasy', 'Science Fiction', 'Animation']
Subset of Reviews: ["Really enjoyed!", "My favourite movie, I will recommend…", …]
Description: ["A smart, funny and original treat, Michael O'Sullivan, The Washington Post, DreamWorks Animations Megamind puts a whole new hilarious twist on the superhero movie.", 'Super villain Megamindś (Will Ferrell) dreams have come true when he conquers the cityś protector Metro Man (Brad Pitt) gaining control of Metro City. But when a new villain (Jonah Hill) is created and chaos runs rampant, the worldś biggest "mind" and his comic sidekick Minion (David Cross) might actually save the day. With an all-star cast - including Tina Fey - and amazing animation, Megamind is packed with high-flying action and non-stop laughs.', "Villainy is a way of life, but if a villain doesn't have a hero to battle, can he still expect the same sense of satisfaction from his evil deeds? When Megamind (Will Ferrell) finally defeats his long-time nemesis Metro Man (Brad Pitt), he ecstatically goes about laying ruin to Metro City, reveling in the fact that he can now have anything he wants. Surprisingly, Megamind's glee quickly turns to dissatisfaction as he realizes that the battle was half the fun and everything comes too easy now. Thus begins a plot to turn the nerdy Hal (Jonah Hill) into Tighten, a new hero for Megamind to battle. What Megamind doesn't count on is that Tighten may not turn out to be as good as he's meant to be. Nor does he anticipate falling in love with his old rival's girl Roxanne Ritchi (Tina Fey). In the end, Megamind finds he must rethink his assumptions about good, bad, and what makes him happy. Megamind and Despicable Me are remarkably similar movies: both deal with villains who are unfulfilled by their evil ways and who, despite their despicable natures, have some essential goodness deep down inside. Both villains are presented in highly stylized animation and the 3-D effects in each film are skillful and effective without being overdone. Megamind has Minion, who watches over him, while Gru has a whole crew of mindless minions supporting his evil ventures. Finally, both films are solidly entertaining. Because the two films were released within a few months of one another, speculation about who copied who will no doubt run rampant, but the bottom line is that both films are well worth seeing. (Ages 7 and older) –Tami Horiuchi"]

General Prompt Guideline for User (S_v)

Prompt : Create a summary from this data about the movie within 100 words

Guideline:

- Identify the main plot points or themes presented in the description.
- Highlight key characters and their roles.
- Summarize the overall premise and main events of the item.
- Note any comparisons, notable features, or unique aspects mentioned.

Generated Item Profile

GPT 4 Turbo: Megamind is a hilarious animated superhero movie that puts a unique twist on the genre. After villain Megamind (Will Ferrell) finally defeats his nemesis, Metro Man (Brad Pitt), he takes control of Metro City but quickly grows bored without a hero to battle. In an attempt to reignite the excitement, he creates a new rival, Tighten (Jonah Hill), but things don't go as planned. With the help of his sidekick Minion (David Cross) and love interest Roxanne Ritchi (Tina Fey), Megamind must reconsider his views on good and evil. Packed with action and laughs, Megamind is a must-watch family film.

LLaMa3-70B: Megamind is an animated superhero comedy that tells the story of a supervillain named Megamind (Will Ferrell) who conquers Metro City's protector, Metro Man (Brad Pitt). Feeling unsatisfied with his easy victory, Megamind creates a new hero, Tighten, and battles him. Along the way, he falls in love with Roxanne Ritchi (Tina Fey), a journalist. The movie explores themes of fulfillment and friendship, with standout characters.

Figure 4: Item profile generation.

enjoy?", whereas for automotive, it could inquire, "What car features are important to the user?".

In generating user profiles, LLMs (e.g., GPT-4 Turbo) are prompted to summarize the types of items that would appeal to the user based on their past purchases and reviews. Item profiles incorporate insights from how different users reviewed the product, with a focus on its unique features. This methodology ensures accurate and insightful summaries, enhancing the performance of GLoM. Figures 3 and 4 depict user and item profile examples from the Amazon-Movies and TV dataset, respectively.

3.2.4. Fixed-size Embeddings

To convert user and item profiles into fixed-size embeddings, we utilize the approach outlined in [18]. The method involves creating a fixed-dimensional vector representation for each profile by applying a model fine-tuned with specific instructions for various tasks. Given a user profile P_u or an item profile P_v, the embeddings E_u and E_v for the user and item profiles are computed as:

$\mathbf{u}_{p_u} = f_e(P_u), \quad \mathbf{v}_{p_v} = f_e(P_v)$. Here, $f_e(\cdot)$ represents the function to transform the text inputs into fixed-size vectors while preserving the contextual information of the generated profiles.

3.3. User-Item Polarity Edge Creation

To refine the relationship between users and items, we introduce the concept of user-item polarity edge creation. This approach is based on the feedback a user u provides about a specific item v. The polarity score p_{uv} is derived from analyzing the review score by combining the user rating with the sentiment expressed in the review text. This score is then used to define a polarity edge as $e_{uv} = (u, v, p_{uv})$, representing the strength and direction of the interaction between the user and the item. These polarity edges are integrated into the KG. To calculate the polarity score, our model utilizes sentiment analysis to extract sentiment scores from reviews, following the method described by Hartmann et al. [19]. It calculates the average sentiment score across all reviews and compares the sentiment of a specific review to this average. If the sentiment score of the user's review for an item is greater than or equal to the average score, it is counted as the user liking the item; otherwise, it is considered the opposite. By combining this sentiment analysis with the user rating, the model provides a more contextual understanding of user feedback, distinguishing between outlier opinions and the general perspective. The model assigns a weight to both the user rating r_v^u and the sentiment score s_v^u as: $p_{uv} = \alpha \times r_v^u + (1 - \alpha) \times s_v^u$. Here, α represents the weight for the user rating, while $(1 - \alpha)$ represents the weight for the sentiment score. This equation balances the influence of the numerical rating and the sentiment expressed in the review, resulting in a more accurate and nuanced assessment of the review's actual value.

3.4. User-User Similarity Edge Generation

We define a user-user similarity edge based on the similarity between users u_a and u_b, denoted as $e_{u_a u_b} = (u_a, u_a, \text{sim}(u_a, u_b))$. The similarity is calculated based on the cosine similarity of their polarity scores p, which measures the angle between vectors representing the polarity scores for items both users have rated. The similarity function between users u_a and u_b is given by:

$$\text{sim}(u_a, u_b) = \frac{\sum_{v \in \text{CM}} \left(p_{u_a,v} - \bar{p}_{u_a}\right)\left(p_{u_b,v} - \bar{p}_{u_b}\right)}{\sqrt{\sum_{v \in \text{CM}} \left(p_{u_a,v} - \bar{p}_{u_a}\right)^2} \sqrt{\sum_{v \in \text{CM}} \left(p_{u_b,v} - \bar{p}_{u_b}\right)^2}} \tag{1}$$

where $\bar{p}_a = \frac{1}{|\text{CM}|} \sum_{v \in \text{CM}} p_{v,a}$ and, $\bar{p}_b = \frac{1}{|\text{CM}|} \sum_{v \in \text{CM}} p_{v,b}$. CM represents the set of items rated by both users u_a and u_b, $p_{u_a,v}$ and $p_{u_b,v}$ are the polarity scores for item v given by users u_a and u_b, and $\bar{p}_{u_a}$ and $\bar{p}_{u_b}$ are the average polarity scores for users u_a and u_b, respectively. If the similarity score between two users exceeds the threshold t_{sim}, it indicates they share similar online buying behavior. This similarity computation leverages both the ratings and the sentiment expressed in the ratings, providing a comprehensive measure of user similarity for collaborative filtering.

3.5. Pre-trained Model (PM)

In the pre-training stage, we aim to design an architecture that can efficiently and jointly reason over text and structured data. The knowledge graph provides rich information and a solid knowledge base for solving the lookalike problem, but they lack language understanding. To ease this we introduce a pre-trained model $f_{\text{pre}}(\cdot; \theta_{\text{pre}})$ parameterized with θ_{pre} which learns graph structure along with generated texts. This

model is designed to generate embeddings of entities $e \in E$ in the knowledge graph G, formulated as $h_e = f_{\text{pre}}(G; \theta_{\text{pre}})$.

We propose two types of entity representations: interaction-based and profile-based. Interaction-based representations capture interaction facts, including polarity and similarity information, while profile-based representations focus on textual and semantic details. These two representations are learned simultaneously in the same vector space without enforcing unification. The pretraining energy function is defined as:$\mathscr{F} = \mathscr{F}_{\mathscr{I}} + \mathscr{F}_{\mathscr{G}}$, where $\mathscr{F}_{\mathscr{I}}$ is the energy function of interaction-based representations defined as: $\mathscr{F}_{\mathscr{I}} = \|\mathbf{u}_i + \mathbf{r} - \mathbf{v}_i\|$ and $\mathscr{F}_{\mathscr{G}}$ is the energy function of user and item profile-based representations [20]. To make the learning process of $\mathscr{F}_{\mathscr{G}}$ compatible with $\mathscr{F}_{\mathscr{I}}$, we define $\mathscr{F}_{\mathscr{G}}$ as:$\mathscr{F}_{\mathscr{G}} = \mathscr{F}_{\mathscr{G}\mathscr{G}} + \mathscr{F}_{\mathscr{G}\mathscr{I}} + \mathscr{F}_{\mathscr{I}\mathscr{G}}$, where $\mathscr{F}_{\mathscr{G}\mathscr{G}} = \|\mathbf{u}_{p_u} + \mathbf{r} - \mathbf{v}_{p_v}\|$, in which head and tail are profile generation-based representations of user and item. Also, we have $\mathscr{F}_{\mathscr{G}\mathscr{I}} = \|\mathbf{u}_{p_u} + \mathbf{r} - \mathbf{v}_i\|$ and $\mathscr{F}_{\mathscr{I}\mathscr{G}} = \|\mathbf{u}_i + \mathbf{r} - \mathbf{v}_{p_v}\|$, where one of $\mathbf{u}$ or $\mathbf{v}$ uses profile generation-based representation and the other uses interaction-based representation. In this paper, we generate representations for user/item profile using the method described in Sec. 3.6. To capture the dynamic intentions of users, we design a link prediction task in the pre-trained stage for learning entity representations, which is defined as:

$$\mathscr{L}_{\text{pre}} = \sum_{(u,r,v)\in\mathscr{E}} \sum_{(u',r,v')\in\mathscr{E}^{-1}} [\gamma + f(u,r,v) - f(u',r,v')]_+ , \tag{2}$$

where (u, r, v) is positive triples, which actually exist in the KG, (u', r, v') is negative triples, γ is the margin, $f(u, r, v)$ is the score function.Here, the score function is defined as $f(u, r, v) = \|\boldsymbol{e}_u + \boldsymbol{e}_r - \boldsymbol{e}_v\|_{1,2}$, where $\boldsymbol{e}_u$, $\boldsymbol{e}_r$ and $\boldsymbol{e}_v$, are the embeddings of head entities, relations and tail entities, respectively. From the other perspective, sum of a head and relation embeddings reaches a near point to the related tail embeddings, therefore it can be regarded as a query to visit tail entities related to the head entity with the relation. In this paper, we term it as *knowledge query*. Later Graph Learning Model (GLM) aggregates the knowledge queries for aligning neighbor node embeddings in the vector space.

3.6. Graph Learning Model (GLM)

After the Pre-trained Model, the embeddings are passed through a Graph Convolution Network for smoothing. Regarding aggregation, GLoM aggregates *knowledge queries* instead of neighboring node features. A knowledge query (for definition refer to Sec. 3.7) from a source node u to a destination node v with a relation r, aggregated during the update of the destination node features, is defined as: $\mathscr{Q}(u, r, v) = \boldsymbol{e}_u + \boldsymbol{e}_r$. Here, $\boldsymbol{e}_u$ and $\boldsymbol{e}_r$ represent embeddings of the source node and the relation, respectively, and the triple (u, r, v) exists within the Knowledge Graph (KG). One of the primary benefits of aggregating knowledge queries is the alignment of node embeddings within the vector space. In the update phase, GLoM combines the aggregated knowledge queries with the target node embedding using linear transformation.

3.6.1. Aggregator

Here, we propose mean and attention-based aggregators. We redefine the notation of knowledge queries for multiple layers of GLoM as: $\mathscr{Q}^{l-1}_{(u,r,v)} = \boldsymbol{h}^{l-1}_u + \boldsymbol{h}^{l-1}_r$, where l indicates the l-th layer, $\mathscr{Q}^{l}_{(u,r,v)}$ is the l-th query from source u to destination v with relation r, and $\boldsymbol{h}^l_u$ and $\boldsymbol{h}^l_r$ represent the l-th embeddings of the source and relation, respectively. Initial embeddings of all nodes and

relations obtained from PM are utilized, i.e., $\boldsymbol{h}^0 u = \boldsymbol{e}_u$, $\boldsymbol{h}^0 r = \boldsymbol{e}_r$. Two types of aggregators are formulated as follows.

Mean aggregator: This aggregator simply computes the average of neighboring knowledge queries:

$$\boldsymbol{m}_{\mathbb{Q}}^l = \text{MEAN}\left(\left\{\mathbb{Q}_{(u,r,v)}^{l-1}, u \in \mathcal{N}(v), r \in \mathcal{R}(u,v)\right\}\right) \tag{3}$$

Here, $\boldsymbol{m}_{\mathbb{Q}}^l$ is the l-th message of knowledge queries, $\mathcal{N}(v)$ is the set of neighbor nodes of node v, $\mathcal{R}(u,v)$ is the set of relations between u and v.

Attention aggregator: Unlike the mean aggregator, the attention aggregator weights each knowledge query based on its importance. Two types of attention aggregators are offered:

$$\boldsymbol{m}_{\mathbb{Q}}^l = \sum_{u \in \mathcal{N}(v)} \alpha_{(u,r,v)} \mathbb{Q}_{(u,r,v)}^{l-1} \tag{4}$$

$$\alpha_{(u,r,v)} = \frac{e_{(u,r,v)}}{\sum_{k \in \mathcal{N}(v)} e_{(k,r,v)}} \tag{5}$$

$$(A1) \quad e_{(u,r,v)} = \left(\mathbb{Q}_{(u,r,v)}^{l-1}\right)^T \boldsymbol{h}_v^{l-1} \tag{6}$$

$$(A2) \quad e_{(u,r,v)} = \text{LeakyReLU}\left(\boldsymbol{a}^T \left(\mathbb{Q}_{(u,r,v)}^{l-1} \parallel \boldsymbol{h}_v^{l-1}\right)\right) \tag{7}$$

where $\alpha_{(u,r,v)}$ is the normalized attention coefficient, $\boldsymbol{a} \in \mathbb{R}^{2H}$ is the trainable parameter, H is the dimension of the embeddings, and $\parallel$ denotes concatenation. The first method (Eq. (6)) employs the inner product between the knowledge query and the destination node to calculate attention coefficients, akin to the attention mechanism of Knowledge Graph Convolutional Networks [21]. The second method (Eq.(7)) introduces trainable parameters that enable automatic adjustment of how knowledge queries are aggregated based on the loss function.

3.6.2. Update

After aggregating the knowledge queries, new embeddings for all nodes and relations are obtained by combining the destination node embedding with the aggregated queries. The update rule is as: $\boldsymbol{h}_v^l = W^l(\boldsymbol{h}v^{l-1} + \boldsymbol{m}^l\mathbb{Q}) + \boldsymbol{b}^l, \boldsymbol{r}^l = W^l\boldsymbol{r}^{l-1} + \boldsymbol{b}^l$. Here, $\boldsymbol{h}_v^l$ and $\boldsymbol{r}^l$ are the updated embeddings for target nodes v and relations, serving as inputs for the next layer. W^l and $\boldsymbol{b}^l$ are the trainable parameters for the l-th layer. This rule combines the destination node embedding with the aggregated queries and applies a linear transformation. The same transformation is applied to relation embeddings, enabling translation between entities and relations. No non-linear functions are used in this process.

3.7. Lookalike Audience Expansion

Our goal here is to obtain user embeddings for GLoM and retrive a target list for each given seed list. Therefore, we employ the following unsupervised loss function for training GLM.

$$\mathscr{L}_{\text{final}} = \sum_{(\mathcal{U},\mathcal{F}) \in \mathcal{P}} \sum_{(\mathcal{U},\mathcal{F}') \in \mathcal{P}^{-1}} [\gamma + f(\boldsymbol{h}_{\mathcal{U}}, \boldsymbol{h}_{\mathcal{F}}) - f(\boldsymbol{h}_{\mathcal{U}}, \boldsymbol{h}_{\mathcal{F}'})]_+ \tag{8}$$

Table 1: Statistics of the experimental datasets.

Models	Tasks	Seeds	Expanded Users
Amazon-Books	175	192,49	1,553,694
Amazon-Movies and TV	218	409,337	1,909,337
Amazon-Electronics	190	98,098	1,117,959
Amazon-Automotive	110	254,542	1,696,580

where $(\mathcal{U}, \mathcal{I}) \in \mathcal{P}$ is the positive pairs of various interactions between users and items, $(\mathcal{U}, \mathcal{I}') \in \mathcal{P}^{-1}$ is the random negative one. We obtain the user embeddings and use these embeddings with a similarity threshold T to filter the closest users of each seed user and generate the new list as target prospecting for each marketing campaign as a final output of GLoM framework.

4. Experiment

To demonstrate that GLoM improves lookalike model performance by utilizing both interactions and generated textual information, we conducted the following experiments to address the specified research questions: (RQ1) Does GLoM outperform other lookalike approaches?; (RQ2) How do different components contribute to the model's performance?; (RQ3) How do different user-item profile generations, based on various LLMs, impact pre-training?; (RQ4) How does GLoM perform with limited seed lists?

4.1. Datasets

We evaluate our model using four Amazon datasets[1]: Books, Movies (TV and Movies), Electronics, and Automotive. These datasets include user ratings and reviews, which we preprocess for lookalike modeling. The details of these four datasets are shown in Table 1. The column 'Tasks' refers to the number of items in each dataset for which we expanded the user base from the seed users. For knowledge graph construction, we follow the method outlined in [22]. For training on the Books, Movies, Electronics, and Automotive datasets, seed:non-seed = positive samples:negative samples = 1:10. Each dataset contains a seed set and a non-seed set for training, as well as a set of expanded users for testing. The set of expanded users consists of actual audiences (positive samples) and other candidate users (negative samples).

4.2. Baselines

We describe various approaches suitable for lookalike audience expansion. We reference a few baselines from the deployed lookalike model in WeChat [2]. The baseline approaches presented here aim to predict users as potential targets for advertising campaigns and can run on a single GPU. Baselines are as follows:

Logistic Regression-based Lookalike Model (LR): An end-to-end Logistic Regression (LR) classifier based on the raw features [15].

[1] https://amazon-reviews-2023.github.io

Table 2: Model Performance on Amazon public datasets. We report the mean results over five runs. The best results are marked with a superscript asterisk (*), and the second-best results are underlined.

	Amazon-Books			Amazon-Movies			Amazon-Electronics			Amazon-Automotive		
Models	Prec.	Recall	Pr-auc	Prec.	Recall	Pr-auc	Prec.	Recall	Pr-auc	Prec.	Recall	Pr-auc
LR	0.364	0.544	0.517	0.320	0.588	0.499	0.228	0.577	0.419	0.421	0.590	0.571
PM	0.459	0.651	0.630	0.390	0.674	0.606	0.314	0.675	0.517	0.501	0.709	0.669
GraphSage	0.431	0.622	0.612	0.344	0.606	0.549	0.301	0.617	0.509	0.483	0.611	0.606
LightGCN	0.431	0.639	0.624	0.354	0.626	0.538	0.310	0.625	0.509	0.496	0.632	0.611
Pinterest	0.453	0.651	0.633	0.392	0.678	0.605	0.314	0.679	0.513	0.504	0.715	0.671
MetaHeac	0.451	0.656	0.644	0.395	0.688	0.610	0.325	0.681	0.540	0.506	0.723	0.693
GLoM(mean)	0.466	<u>0.674</u>	0.646	0.425*	0.697	0.627	0.337	0.727*	<u>0.544</u>	0.534	<u>0.745</u>	0.732
GLoM(A1)	0.491*	0.698*	0.661*	<u>0.423</u>	<u>0.707</u>	0.646*	0.365*	0.727*	0.556	0.553*	0.759*	<u>0.745</u>
GLoM(A2)	<u>0.481</u>	<u>0.674</u>	<u>0.654</u>	0.420	0.710*	<u>0.640</u>	<u>0.361</u>	<u>0.713</u>	0.557*	<u>0.549</u>	0.741	0.748*

Table 3: Ablation Study on all Datasets. U/I Profile denotes User/Item Profile Generation, and U/I Polarity denotes User/Item Polarity Edge Generation, U/U Similarity denotes user-user Similarity and PM denotes Pre-training Model. The best results are marked with a superscript asterisk (*).

Models	Amazon-Books		Amazon-Movies		Amazon-Electronics		Amazon-Automotive	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
GLoM w/o U/I Profile	0.429	0.634	0.356	0.612	0.304	0.617	0.479	0.613
GLoM w/o U/I Polarity	0.482	0.675	0.416	0.673	0.324	0.696	0.514	0.711
GLoM w/o U/U Similarity	0.471	0.666	0.356	0.612	0.304	0.603	0.499	0.698
GLom w/o PM	0.431	0.622	0.344	0.606	0.298	0.589	0.465	0.601
GLoM	0.491*	0.698*	0.423*	0.707*	0.365*	0.727*	0.553*	0.759*

Pre-trained Model (PM): We utilize the user embeddings from the PM (please refer to Sec. 3.2) model for retrieving the potential users for our experimental datasets.
GraphSAGE: GraphSAGE is a graph-based learning model that generates node embeddings by sampling and aggregating features from a nodes local neighborhood [6]. **LightGCN:** LightGCN is a simplified Graph Convolutional Network (GCN) model tailored for recommendation tasks [7]. An end-to-end LightGCN is used as baseline similar to GraphSage as GLoM is designed based on graph model.
Pinterest: Pinterest's two-stage approach is employed as one of our baselines [16]. In the first stage, a global user embedding model is trained to create user embeddings. In the second stage, an embedding-based scoring model is used to compute an affinity score for each user in relation to a specific campaign or task.
MetaHeac: It is a state-of-the-art audience expansion model for advertising and has been deployed in WeChat [2]. In their paper, they also included LR and Pinterest as baselines.

4.3. Evaluation Protocols and Model settings

The models run on a single GPU NVIDIA Tesla V100. We employ grid search to fine-tune the hyperparameters, ensuring optimal performance. For the embeddings dimensionality d, we consider options from {25, 50, 100, 150, 200}, while the learning rate α is chosen from {0.001, 0.01, 0.05, 0.1}, and the margin γ is selected from {1, 5, 10}. To generate accurate user and item profiles, we leverage advanced models including GPT (3.5/4 Turbo), Gemini-1.0-pro, and LLaMa 3-70B, provided by OpenAI, Google, and Meta, respectively. We implement our method

LLMs	GLoM			
	Books	Movies	Electronics	Automotive
Gemini-1.0-pro	0.565	0.625	0.619	0.648
GPT 3.5 Turbo	0.690	0.689	0.711	0.747
GPT 4 Turbo	0.696*	0.707*	0.727*	0.759*
LLaMa 3-70B	0.678	0.701	0.724	0.724

Table 4: Comparison on major LLMs using Recall.

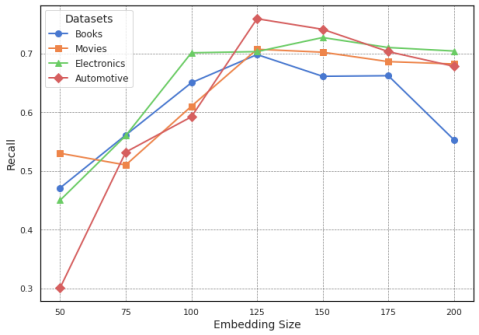

Figure 5: GLoM Performance Across Datasets with Varying Embedding Sizes.

and baselines using PyTorch 1.8.2 in a Python 3.6 environment, leveraging both PyTorch and the Deep Graph Library [23] for baseline implementations.

4.4. Performance Comparison (RQ1)

In this experiment, we evaluate the end-to-end performance of all models using Precision, Recall, and PR-AUC. The actual users who purchased an item are treated as positive examples, while the remaining candidate users are treated as negative examples. Table 2 presents the performance of GLoM compared to other baselines.

GLoM operates as a two-stage method, with the first stage focusing on pre-training. Notably, the pre-trained embeddings (PM) outperformed several baseline models and demonstrated competitive performance against SOTA lookalike model MeatHeac. This highlights the effectiveness of embedding pre-training in enhancing look-alike modeling performance.

Key observations from the evaluation are as follows: (1) GLoM consistently achieved superior performance compared to baseline models, providing strong evidence of its effectiveness. Specifically, GLoM (A1) delivered the best results on four public datasets. (2) Compared to the state-of-the-art lookalike methods such as MetaHeac and Pinterest, GLoM delivers stronger results. This confirms that GLoM improves user targeting by capturing semantic information related to user behavior, sentiments, and preferences while preserving the graph structure. Specifically, GLoM (A1) achieved improvements of 6.40%, 2.76%, 6.75%, and 4.98% compared to the MetaHeac model on the Books, Movies, Electronics, and Automotive datasets, respectively and also significantly outperforming Pinterest. (3) GLoM outperforms ID-based models in performance. Models like LightGCN and GraphSage rely heavily on ID-based information, which may overlook valuable data such as the rich textual information associated with users and items. This indicates that GLoM's learned representations effectively capture global collaborative relationships, going beyond the limitations of ID-based representation techniques. (4) MetaHeac also performed well in certain cases, such as for Amazon-Electronics. As reported in the paper [2], it also achieved better performance than the Pinterest on Amazon datasets as well. (5) GLoM (mean) and GLoM (A1) exhibited more stable performance compared to other methods, with GLoMs (A1) achieved the best results. However, the performance of GLoM (A2) was less stable, potentially due to imbalances in knowledge queries per node. The attention mechanism in GLoM

is guided by the pre-training models loss function, which presents challenges in designing a specialized attention mechanism for this task.
Figure 5 presents the plotted data for four datasets (Books, Movies, Electronics, and Automotive), showing recall performance as the embedding size (dimensions) increases. Note that we plot the embedding size versus recall values specifically for GLoM (A1). As the embedding dimensions grow, recall improves across all datasets at different rates. For most datasets, an embedding size of 125 performs well.Please note that increasing the embedding size can slow down the expansion process during real-time marketing campaigns. However, GLoM can efficiently expand the seed list for the datasets in Table 1 in less than 20 minutes using a single GPU, making it highly effective for rapid campaign scaling.

4.5. Ablation Study (RQ2)

In this paper, we argue that an effective lookalike model needs to better capture both textual signals and graph structures. We analyze the effects of user/item profile generation, polarity edges, user-user similarity, and pre-trained embeddings within our proposed model, GLoM (specifically GLoM-A1). To evaluate these components, we first conduct an ablation study to verify the effectiveness of each module in the Pre-trained Model (PM) and assess PM's overall contribution to GLoM.

We introduce three components in the Pre-trained Model stage, derived from LLM and raw data: User/Item Profile Generation (Sections 3.2 and 3.3), User/Item Polarity Edge Creation (Section 3.5), and User-User Similarity Edge (Section 3.6). First, we remove the User/Item Profile Generation (denoted as GLoM w/o U/I Profile). Second, we remove the User/Item Polarity Edge (denoted as GLoM w/o U/I Polarity). Third, we remove the User-User Similarity Edge (denoted as GLoM w/o U/U Similarity). Lastly, we evaluate the performance of GLoM without the Pre-trained Model. The results are shown in Table 3. We observe the following key findings: 1) The performance of GLoM w/o U/I Profile is the worst compared to GLoM w/o U/I Polarity and GLoM w/o U/U, indicating that the User/Item Profile Generation using LLMs effectively captures semantic signals and user behavior trends, leading to improved performance. 2) Among GLoM w/o U/I Polarity and GLoM w/o U/U, the latter proves to be a more critical component for GLoM. 3) GLoM performs better than GLoM w/o PM, demonstrating that the pre-training phase generates meaningful user and item embeddings that enhance lookalike model performance. Overall, the results in Table 3 highlight that all the proposed components are crucial for constructing an effective lookalike model.

4.6. User-Item Profile Generations using Different LLMs (RQ3)

In this section, we explore the use of different LLMs, such as LLaMa 3-70B, GPT 3.5 and 4 Turbo, and Gemini-1.0-pro , for generating user and item profiles in GLoM. By leveraging these models, we aim to capture deeper semantic relationships between users and items. Table 4 shows the performance of GLoM when using various large language models for user/item profile generation. We observed that GPT-4 Turbo performed exceptionally well in GLoM, particularly in capturing textual signals, as reflected in the results achieved by GLoM (A1) (Table 4). Its ability to model complex user behaviors and preferences from textual data led to significant performance gains. In contrast, LLaMa demonstrated slightly lower performance compared to GPT-4. Gemini-1.0-pro,

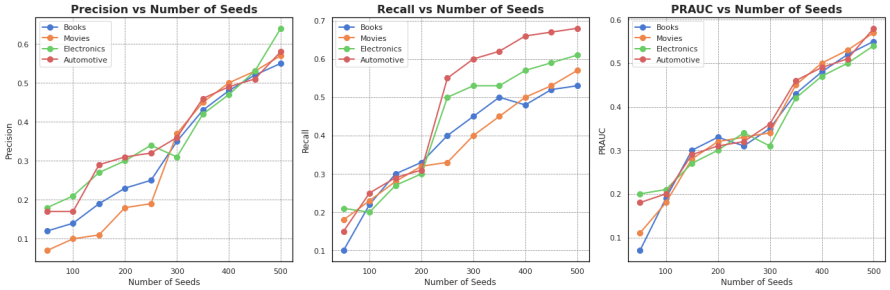

Figure 6: GLoM's Dependency on Seed List Size for Lookalike Modeling.

on the other hand, showed a slightly lower performance than both GPT and LLaMa. Since some users/items have long reviews, and Gemini-1.0-pro is known for handling shorter texts better, this could be a contributing factor. These findings highlight the critical role that model selection plays in optimizing user-item profile generation within GLoM, making it essential to choose the right LLM based on the task at hand.

4.7. Limitations (RQ4)

In lookalike modeling, seed lists serve as the foundation from which the model identifies and expands to find similar users. However, one key limitation is the model's heavy dependence on the quality, representativeness, and size of the seed list. To generalize effectively and ensure diversity, we intentionally minimize the number of cold users (those with limited data) in the seed list. If the seed list contains a significant number of cold users, the performance of the GLoM model is expected to degrade. We conducted experiments to assess GLoMs performance across different seed list sizes to understand its dependencies. Figure 6 shows GLoMs performance on various metrics with varying seed sizes. The results indicate that GLoMs performance drops when the seed list contains fewer than 300 users. However, its performance stabilizes once the seed list exceeds 500 users. Moreover, when dealing with complex user-item interactions or reasoning about user preferenceschallenges for many lookalike modelsGLoM effectively addresses these issues by leveraging the graph structure.

5. Conclusion

In this paper, we address the lookalike problem in advertising platforms by introducing a novel method called the Graph Lookalike Model (GLoM), which leverages the power of Large Language Models. Our model is likely the first model to successfully integrate LLMs with a graph structure, capturing both textual and structural information without using sensitive private data. This enables a deeper understanding of users' behaviors and preferences to identify similar users for advertising campaigns. We demonstrate its effectiveness using public Amazon datasets, which provide the key features required for a lookalike model. Given GLoM's ability to better understand users, it has the potential for application in other industrial scenarios, such as recommendation systems, financial risk analysis, wealth management, and more.

References

[1] B. Kanagal, A. Ahmed, S. Pandey, V. Josifovski, L. Garcia-Pueyo, J. Yuan, Focused matrix factorization for audience selection in display advertising, in: 2013 IEEE 29th International Conference on Data Engineering (ICDE), IEEE, 2013, pp. 386–397.

[2] Y. Zhu, Y. Liu, R. Xie, F. Zhuang, X. Hao, K. Ge, X. Zhang, L. Lin, J. Cao, Learning to expand audience via meta hybrid experts and critics for recommendation and advertising, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 4005–4013.

[3] H. Liu, D. Pardoe, K. Liu, M. Thakur, F. Cao, C. Li, Audience expansion for online social network advertising, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 165–174.

[4] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, Deepfm: a factorization-machine based neural network for ctr prediction, arXiv preprint arXiv:1703.04247 (2017).

[5] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM computing surveys (CSUR) 52 (2019) 1–38.

[6] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

[7] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.

[8] Q. Ma, E. Wagh, J. Wen, Z. Xia, R. Ormandi, D. Chen, Score look-alike audiences, in: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), IEEE, 2016, pp. 647–654.

[9] Q. Ma, M. Wen, Z. Xia, D. Chen, A sub-linear, massive-scale look-alike audience extension system a massive-scale look-alike audience extension, in: Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, PMLR, 2016, pp. 51–67.

[10] A. Ramesh, A. Teredesai, A. Bindra, S. Pokuri, K. Uppala, Audience segment expansion using distributed in-database k-means clustering, in: Proceedings of the Seventh International Workshop on Data Mining for Online Advertising, 2013, pp. 1–9.

[11] G. Y.-Y. Chan, T. Mai, A. B. Rao, R. A. Rossi, F. Du, C. T. Silva, J. Freire, Interactive audience expansion on large scale online visitor data, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2621–2631.

[12] L. Lin, X. Chen, K. Xia, S. Wang, D. Zhang, T. He, Hierarchical information propagation and aggregation in disentangled graph networks for audience expansion, in: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 4702–4709.

[13] J. Shen, S. C. Geyik, A. Dasdan, Effective audience extension in online advertising, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery

and Data Mining, 2015, pp. 2099–2108.

[14] D. Li, Y. Yang, Y.-Z. Song, T. M. Hospedales, Learning to generalize: Meta-learning for domain generalization, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[15] Y. Qu, J. Wang, Y. Sun, H. M. Holtan, Systems and methods for generating expanded user segments, 2014. US Patent 8,655,695.

[16] S. Dewet, J. Ou, Finding users who act alike: transfer learning for expanding advertiser audiences, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2251–2259.

[17] M. M. Rahman, D. Kikuta, S. Abrol, Y. Hirate, T. Suzumura, P. Loyola, T. Ebisu, M. Kondapaka, Exploring 360-degree view of customers for lookalike modeling, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 3400–3404.

[18] H. Su, W. Shi, J. Kasai, Y. Wang, Y. Hu, M. Ostendorf, W.-t. Yih, N. A. Smith, L. Zettlemoyer, T. Yu, One embedder, any task: Instruction-finetuned text embeddings, arXiv preprint arXiv:2212.09741 (2022).

[19] J. Hartmann, M. Heitmann, C. Siebert, C. Schamp, More than a feeling: Accuracy and application of sentiment analysis, International Journal of Research in Marketing 40 (2023) 75–87. URL: https://www.sciencedirect.com/science/article/pii/S0167811622000477. doi:https://doi.org/10.1016/j.ijresmar.2022.05.005.

[20] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[21] H. Wang, M. Zhao, X. Xie, W. Li, M. Guo, Knowledge graph convolutional networks for recommender systems, in: The World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 33073313. URL: https://doi.org/10.1145/3308558.3313417. doi:10.1145/3308558.3313417.

[22] Y. Wang, Q. Xie, M. Tang, L. Li, J. Yuan, Y. Liu, Amazon-kg: A knowledge graph enhanced cross-domain recommendation dataset, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 123–130.

[23] M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, M. Li, J. Zhou, Q. Huang, C. Ma, Z. Huang, Q. Guo, H. Zhang, H. Lin, J. Zhao, J. Li, A. J. Smola, Z. Zhang, Deep graph library: Towards efficient and scalable deep learning on graphs, CoRR abs/1909.01315 (2019). URL: http://arxiv.org/abs/1909.01315. arXiv:1909.01315.