International Journal of Electrical and Computer Engineering (IJECE)
Vol. 16, No. 3, June 2026, pp. 1518~1530
ISSN: 2088-8708, DOI: 10.11591/ijece.v16i3.pp1518-1530

A survey of retrieval algorithms in ad and content recommendation systems

Yu Zhao[1], Fang Liu[2], Yuan Yuan[3], Yifan Dang[4]
[1]Rotman School of Management, University of Toronto, Toronto, Canada
[2]Yale School of Management, Yale University, New Haven, United States of America
[3]School of Engineering and Science, Stevens Institute of Technology, Hoboken, United States of America
[4]Robert H. Smith School of Business, University of Maryland, College Park, United States of America

Article Info

Article history:

Received Feb 11, 2026
Revised Mar 28, 2026
Accepted Apr 27, 2026

Keywords:

Ad targeting
Industrial recommendation systems
Information retrieval
Large language models
Recommendation systems
Retrieval algorithms
Two-tower neural network

ABSTRACT

This paper presents a survey of retrieval algorithms used in advertising recommendation and organic content recommendation systems. Modern digital platforms rely on retrieval-based models to efficiently match users with relevant advertisements or personalized content. This survey reviews key techniques including inverted index methods, collaborative filtering, content-based filtering, hybrid recommendation models, and the two-tower neural network architecture widely used in large-scale recommendation systems. The paper compares the objectives, data utilization strategies, and evaluation metrics of ad targeting and organic retrieval systems. Practical challenges such as cold-start problems, data quality, scalability, and privacy considerations are also discussed. This survey further highlights the growing connection between industrial recommendation pipelines and emerging retrieval mechanisms used in large language model (LLM) systems. This survey provides insights into the design principles of modern retrieval systems and outlines future research directions at the intersection of recommendation systems and LLM.

Corresponding Author:

Fang Liu
Yale School of Management, Yale University
New Haven, CT, United States of America
Email: fangliu435@gmail.com

1. INTRODUCTION

With the rapid growth of digital platforms, delivering personalized content and advertisements has become essential for improving user engagement and platform revenue. Retrieval algorithms play a central role in recommendation systems by matching users with relevant advertisements and organic content [1], [2]. Ad targeting focuses on maximizing engagement and conversion rates through personalized advertisements based on user profiles and behavioral data, whereas organic retrieval systems aim to improve user experience by recommending content aligned with users' interests and preferences.

In recent years, recommendation systems have evolved from traditional information retrieval techniques to deep learning–based retrieval architectures used in large-scale industrial platforms [3], [4], with additional recent surveys [5]. Various recommendation approaches have been developed to improve accuracy and scalability, including content-based filtering, collaborative filtering, and hybrid recommendation methods [6], [7]. More recently, neural retrieval architectures such as the two-tower model have been widely adopted in industrial recommendation pipelines due to their effectiveness in large-scale candidate generation and representation learning.

This paper surveys retrieval algorithms used in advertising and organic recommendation systems, covering representative approaches such as content-based filtering, collaborative filtering, hybrid models, and neural retrieval architectures including the two-tower model. Practical challenges such as cold start, scalability, data quality, and privacy concerns are also discussed.

Unlike many existing surveys that broadly review recommendation algorithms, this paper focuses specifically on retrieval-stage algorithms in modern industrial recommendation pipelines. The survey examines retrieval techniques used in both advertising and organic recommendation systems and highlights how candidate generation operates in large-scale platforms. It also discusses potential connections between retrieval-based recommendation systems and large language model (LLM)–enhanced retrieval architectures.

Although this survey does not explore this topic in depth, the retrieval mechanisms discussed here share conceptual similarities with retrieval-augmented generation (RAG) frameworks used in LLM [8]–[13], where retrieval serves as an intermediate step to identify relevant information prior to downstream generation or ranking, thereby improving relevance, contextual grounding, and overall system performance. In RAG systems, relevant information is first retrieved from external knowledge sources and then incorporated into language models to improve the quality and factual grounding of generated responses. Similarly, recommendation systems rely on retrieval algorithms to identify relevant user and item representations before downstream ranking or prediction tasks.

Overall, this survey provides a structured overview of retrieval algorithms with a particular focus on industrial deployment scenarios. By analyzing both traditional retrieval techniques and modern neural architectures, the paper highlights key design principles and system challenges in large-scale industrial recommendation pipelines.

The remainder of this paper is organized as follows. Section 2 introduces major ad targeting techniques and indexing strategies. Section 3 reviews retrieval architectures used in recommendation systems, including the two-tower model. Section 4 compares advertising recommendation and organic retrieval pipelines. Section 5 discusses evaluation metrics and experimentation frameworks, followed by the conclusion in section 6.

In this survey, representative retrieval algorithms used in advertising and content recommendation systems are reviewed. Relevant studies were identified through major academic databases such as IEEE Xplore, ACM Digital Library, and Google Scholar, focusing on publications from 2015 to 2024. Selection criteria included relevance to retrieval-stage recommendation systems, citation impact, and industrial applicability. Studies were selected based on their influence in both academic research and industrial practice. The reviewed methods are categorized into traditional retrieval techniques and neural retrieval architectures, and each method is analyzed in terms of its mechanism, scalability, industrial applicability, and usage scenarios in modern recommendation pipelines.

The main contributions of this paper are summarized as follows:

a. This paper provides a focused survey on retrieval-stage algorithms in modern recommendation systems, which are less emphasized in existing literature.
b. It presents a comparative analysis between ad targeting systems and organic recommendation systems from a unified retrieval perspective.
c. It analyzes trade-offs among different retrieval approaches in industrial deployment scenarios.
d. It discusses emerging connections between recommendation retrieval and LLM-based retrieval frameworks.

2. AD TARGETING MODELS

Ad targeting is widely used in modern digital marketing to deliver personalized advertisements to specific audiences and maximize engagement and conversion rates. One common technique is the use of inverted indexes, which efficiently match user profiles with relevant advertisements in large-scale information retrieval systems [14]–[16]. Several targeting strategies are commonly used, including demographic targeting, retargeting, keyword targeting, behavioral targeting, and contextual targeting [17], [18].

2.1. Inverted index

Ad targeting commonly leverages inverted indexes to improve the efficiency and accuracy of delivering personalized advertisements [14]. An inverted index is a data structure that maps items (such as advertisements) to associated keywords or attributes, enabling fast search and retrieval. As illustrated in Figure 1, the inverted index links keywords to corresponding items, allowing the system to quickly retrieve relevant advertisements based on user queries or profile attributes. In ad targeting systems, the inverted index typically operates through several stages. First, an index is constructed from available advertisements, where each advertisement is represented by a set of keywords or attributes describing its content and target audience. Next, users are profiled based on their online activities, such as search queries, browsing history, social media interactions, and purchase behavior. Each user profile is represented by a set of keywords or attributes reflecting the user's interests and preferences.

Finally, when a user interacts with a platform, the system retrieves the user profile and matches it against the inverted index. By locating keywords associated with the user, the system can efficiently retrieve advertisements with matching or related attributes. This mechanism enables efficient real-time ad targeting by allowing platforms to quickly retrieve advertisements that match users' interests even in large-scale datasets.

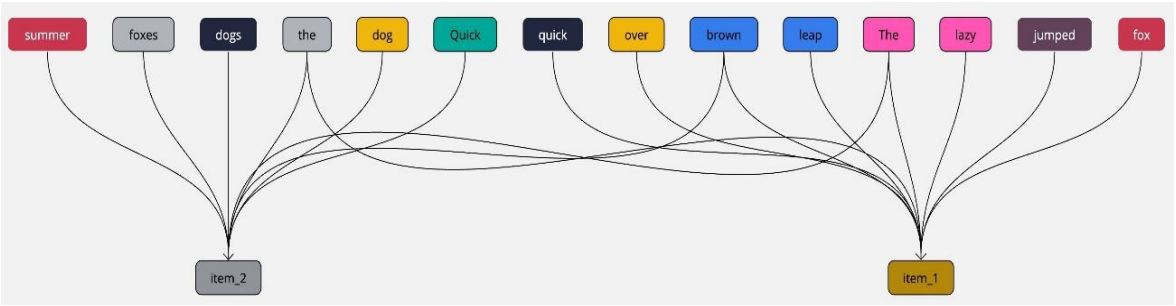

Figure 1. Inverted index structure for advertisement retrieval

2.2. Demographic

Demographic targeting delivers advertisements to users based on demographic attributes such as age and gender. For example, toys are primarily marketed to children, educational resources to teenagers, and retirement or financial planning services to older users. Demographic information is typically obtained from user registration data, profile information, or inferred from behavioral signals such as browsing history and content consumption patterns. These attributes are then incorporated into the advertisement indexing process, where advertisements are associated with demographic-related keywords or attributes. During the retrieval process, the system compares user demographic characteristics with the indexed advertisements. By matching user attributes with relevant demographic categories, the advertising system can efficiently retrieve advertisements that are more likely to be relevant to the user [17], [18].

2.3. Re-targeting

Retargeting, also known as remarketing, focuses on users who have previously interacted with a website or application but have not completed a desired action, such as making a purchase. This technique aims to re-engage potential customers by reminding them of products or services they have previously viewed. As illustrated in Figure 2, the retargeting process generally involves collecting user interaction data, tracking browsing behavior, and delivering targeted advertisements on external platforms such as social media or search engines to encourage users to return and complete the desired action.

Figure 2. Retargeting workflow in digital advertising systems

In practice, retargeting systems rely on several stages of data processing. First, user interaction data is collected through cookies, tracking pixels, and other tracking technologies. This information may include visited pages, viewed products, and user actions on the platform. Next, advertisements are indexed according to the specific actions performed by users. For instance, a product left in a shopping cart may trigger advertisements associated with that product category or related items. Finally, when the user returns to the website or continues browsing other platforms, the system matches the user's previous interactions with relevant advertisements, increasing the likelihood that the user will return and complete the transaction [17], [18].

2.4. Keyword targeting

Keyword targeting delivers advertisements based on keywords that users enter in search queries or that appear in the content they are currently viewing. By analyzing these textual signals, advertising systems can match user intent with relevant advertisements.

Recent studies suggest that LLMs can further enhance keyword targeting by expanding and refining keyword sets. Models such as GPT-4 [10] and other transformer-based architectures [9], [19] are capable of generating related keyword variations that capture different ways users express their search intent. For example, a keyword such as "running shoes" may be expanded into related phrases including "athletic footwear," "jogging sneakers," or "marathon trainers." Such semantic expansion increases the likelihood of matching relevant queries and improves advertisement relevance.

In practice, keywords are typically collected from search queries, webpage content, and user-generated text such as comments or reviews. Advertisements are indexed using these keywords so they can be efficiently retrieved when related queries occur. When users search for or interact with relevant content, the system compares the associated keywords with indexed advertisements and retrieves ads that match the user's current context or search intent.

2.5. Behavioral targeting

Behavioral targeting is a widely used technique in advertisement recommendation systems. It relies on analyzing users' online activities in order to deliver advertisements that reflect their interests and preferences. User behavioral data collected from browsing history, search queries, social media interactions, and purchase behavior are aggregated over time to build user profiles that capture long-term interests and engagement patterns.

Advertisements are then indexed according to behavioral patterns. For example, advertisements related to outdoor equipment may be associated with behavioral indicators such as hiking, camping, or adventure-related searches. During the retrieval process, the system analyzes user behavior and matches it with advertisements that correspond to these observed interests. For instance, users who frequently search for hiking gear may be presented with advertisements for outdoor equipment [17], [18].

2.6. Contextual targeting

Contextual targeting delivers advertisements based on the content of the webpage that a user is currently viewing. Instead of relying primarily on user profiles, this approach analyzes the surrounding content to determine which advertisements are most relevant in a given context.

Contextual information is typically extracted from webpage text, images, and metadata. Natural language processing (NLP) and content analysis techniques are commonly used to understand the semantic meaning of the page and identify relevant topics [19], [20].

Advertisements are indexed with keywords that correspond to specific contextual themes. For example, advertisements for gym memberships or sportswear may be associated with keywords related to fitness, health, and exercise. When a user visits a webpage, the system analyzes the page content and retrieves advertisements that match the detected context [18].

3. RETRIEVAL SYSTEMS

Recommendation systems have been widely studied in the literature [1], [20]. Ad targeting focuses on delivering personalized advertisements to users based on their profiles and behaviors, aiming to maximize engagement and conversion rates. In contrast, organic retrieval emphasizes improving user experience by recommending content or products that align with users' preferences without direct monetary influence.

3.1. Organic retrieval

Organic retrieval, also known as organic recommendation, aims to enhance user experience by providing personalized content or product suggestions based on user preferences and behavior. These systems analyze historical interactions to identify items that are most relevant to each user.

Organic retrieval systems are widely used in domains such as e-commerce, streaming platforms, and social media [1], [20]. Large-scale industrial systems such as YouTube rely heavily on neural retrieval architectures for candidate generation [21]. For example, Netflix recommends movies based on users' viewing history, while Spotify's Discover Weekly introduces new music according to listening behavior.

Despite their success, organic retrieval systems face several challenges, including data sparsity, the cold start problem, and maintaining recommendation diversity [22]. Data sparsity occurs when interaction data is limited, while the cold start problem arises when new users or items lack historical records. Maintaining diversity is also important to prevent recommendations from becoming overly narrow.

Traditional organic retrieval methods include content-based filtering, collaborative filtering, and hybrid approaches. Content-based filtering recommends items with similar characteristics to those previously preferred by the user. Collaborative filtering identifies patterns among users with similar preferences and has evolved into neural collaborative filtering architectures for large-scale systems [7], [23]. Many modern platforms combine both approaches to improve recommendation accuracy and robustness. To provide a clearer overview of different recommendation approaches, Table 1 summarizes the key characteristics, advantages, and limitations of several commonly used retrieval methods. As shown in Table 1, neural retrieval methods provide better scalability but require large-scale training data.

Table 1. Comparison of major recommendation retrieval approaches

Method	Key Idea	Strength	Limitation
Content-based filtering	Recommends items with similar attributes	Simple and interpretable	Limited discovery of new interests
Collaborative filtering	Uses behavior of similar users	Captures implicit preferences	Cold start and sparsity issues
Hybrid methods	Combines multiple recommendation strategies	Improves accuracy and robustness	Higher system complexity
Neural retrieval (two-tower)	Learns user/item embeddings for similarity search	Scalable for large systems	Requires large training datasets

3.2. Two-tower network for retrieval

The two-tower model, also known as the dual-tower model, is a deep learning architecture widely used in recommendation systems for content retrieval [3], [4], [21], [24], [25]. It extends traditional matrix factorization approaches [6] by learning embedding representations for users and items through neural networks.

The architecture consists of two independent neural networks: a user tower and an item tower. The user tower encodes user features such as demographic attributes, browsing history, and past interactions into a latent representation. The item tower encodes item features such as metadata, textual descriptions, or visual content into item embeddings. Both representations are projected into a shared latent space where similarity between users and items can be computed.

Let x_u denote the input features of a user and x_i denote the input features of an item. The user tower maps the user features into a latent representation u through a neural mapping function $f(\cdot)$, while the item tower maps item features into an embedding vector v through another neural function $g(\cdot)$:

$$u = f(x_u)$$

$$v = g(x_i)$$

The relevance between users and items is typically measured using similarity functions such as the dot product:

$$s(u, v) = u^T v$$

or cosine similarity

$$s(u, v) = \frac{u^T v}{||u||\ ||v||}$$

As illustrated in Figure 3, this architecture allows user and item representations to be learned independently while enabling efficient large-scale retrieval.

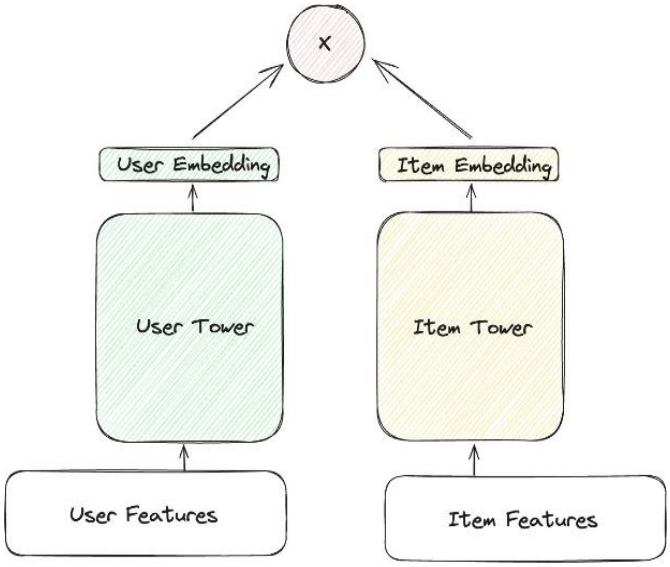

Figure 3. Architecture of the two-tower neural retrieval model

3.3. Training and inference

Training the two-tower model involves learning embeddings that capture compatibility between users and items. During training, user–item pairs are sampled from historical interaction data. Positive pairs represent observed interactions, while negative samples represent non-interactions. This strategy enables the model to distinguish relevant items from irrelevant ones. Negative sampling techniques are often applied to address the imbalance between positive and negative examples [11], [26].

The model is typically optimized using loss functions such as cross-entropy loss [23], [27], [28] or pairwise ranking loss, which are commonly used in implicit feedback recommendation systems. Model parameters are updated through gradient-based optimization algorithms such as stochastic gradient descent (SGD) or Adam.

During inference, the trained model generates embedding vectors for users and items. The user tower produces a vector representation for a given user, while the item tower generates embeddings for candidate items. The similarity between user and item vectors is then computed using similarity metrics such as dot products or cosine similarity. The system retrieves the top-N items with the highest similarity scores and recommends them to the user. In large-scale industrial systems, this retrieval process is often accelerated using efficient indexing techniques such as approximate nearest neighbor (ANN) search and distributed retrieval infrastructure [3], [16].

The overall retrieval procedure of a two-tower recommendation system can be summarized as follows. First, the user tower encodes user features to generate a user embedding vector. Second, the item tower produces embedding representations for candidate items, which are often precomputed and stored in an indexing structure. Third, a similarity function such as dot product or cosine similarity is used to measure the relevance between the user embedding and item embeddings. Finally, the system retrieves the top-N items with the highest similarity scores and returns them as recommendation candidates.

3.4. Challenges and extensions

Modern recommendation systems employ a variety of retrieval models beyond the two-tower architecture. Dense vector retrieval methods are widely adopted in large-scale systems [16], where user and item representations are learned in a shared embedding space and efficient nearest-neighbor search is used for candidate generation.

In addition, graph neural network (GNN)–based models have been introduced to capture complex user–item relationships through interaction graphs [29]–[31]. Representative extensions include social recommendation models [32] and lightweight graph convolution approaches such as LightGCN [33], while survey studies provide broader overviews of this research area [34]–[36].

Knowledge graph–based recommendation methods further enhance retrieval by incorporating structured relational information between users and items, improving both accuracy and explainability in complex recommendation scenarios [37]–[39].

Multimodal retrieval approaches further enhance recommendation performance by integrating information from multiple modalities such as text, images, and audio [22], [40]. A key challenge in recommendation systems is the cold start problem [41]–[43], where new users or items lack sufficient interaction data. To address this issue, several strategies have been explored, including transfer learning, which leverages knowledge from large-scale datasets, and synthetic data generation, which augments sparse interaction data. Federated learning has also been proposed to enable collaborative model training across distributed data sources while preserving user privacy [44].

Another important factor affecting system performance is data quality [42]. Recommendation effectiveness relies heavily on reliable interaction data, and noisy or sparse datasets can significantly degrade model performance. Recent work has proposed several extensions to enhance the traditional two-tower architecture. Multi-task learning allows the model to jointly optimize multiple objectives such as click-through rate prediction and user engagement [26], [45]. The three-tower architecture further extends this design by introducing an additional network to model contextual information, including temporal signals, location data, and cross-feature interactions [46], [47].

As illustrated in Figure 4, the three-tower architecture processes user features, item features, and contextual features through separate neural towers before combining them in a shared representation layer for final prediction. This design enables the model to capture richer interactions and improve recommendation performance in complex scenarios.

Recent studies have also explored integrating LLMs into recommendation pipelines [11]–[13]. LLMs can generate synthetic interaction data and enhance semantic representations of textual features, helping alleviate cold start issues and improving the robustness of recommendation systems. In addition, LLMs can be used to enrich user, item, and contextual representations, complementing multi-tower architectures and improving recommendation performance in sparse and complex data scenarios.

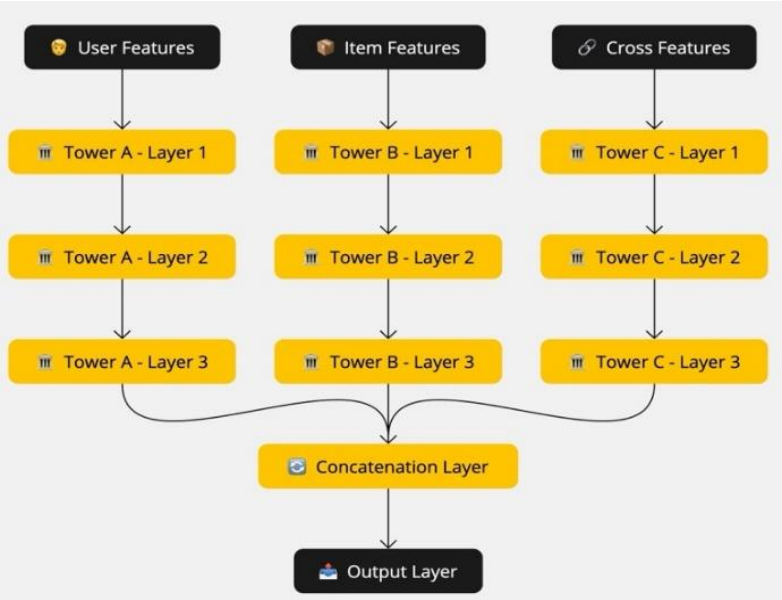

Figure 4. Architecture of the three-tower neural recommendation model

4. COMPARING AD TARGETING AND ORGANIC RETRIEVAL

From a system design perspective, both paradigms share similar retrieval mechanisms but diverge in optimization objectives and evaluation criteria. Modern advertising systems often rely on large-scale click prediction models [45]. At a fundamental level, ad targeting and organic recommendation retrieval share a similar objective: matching user intent with relevant content. Advertising systems analyze user behavior and demographic information to predict which advertisements are most likely to generate engagement, while search engines and recommendation systems analyze queries, interaction histories, and content features to identify relevant items. In both cases, machine learning and large-scale data processing play a central role in improving the accuracy of content delivery. This methodological similarity highlights the increasing convergence between targeted advertising technologies and content retrieval systems [1], [20].

Index-based ad targeting is conceptually similar to the inverted index widely used in organic information retrieval [15]. Both approaches rely on structured indexing mechanisms to efficiently match users with relevant content. In advertising systems, user profiles are categorized based on behavioral patterns, interests, and demographic attributes, allowing advertisers to identify potential audiences for specific campaigns. In organic retrieval systems, inverted indices map keywords or features to documents or items, enabling search engines to rapidly retrieve relevant results in response to user queries.

Targeting strategies in advertising also resemble retrieval mechanisms used in organic search systems. Interest-based targeting displays advertisements to users according to their inferred interests and browsing behaviors, which parallels recommendation systems that retrieve content aligned with user preferences. Keyword targeting focuses on specific words or phrases and functions similarly to traditional search retrieval, where user queries are matched against indexed content containing relevant terms. Despite these similarities, the two systems differ in their optimization goals, data usage, and evaluation metrics. Table 2 summarizes the key differences between ad targeting systems and organic retrieval systems in terms of objectives, data sources, optimization targets, and evaluation metrics. These differences are further reflected in their data usage and evaluation strategies. Ad targeting relies heavily on demographic and behavioral data for precise targeting, while organic retrieval emphasizes historical interactions to improve relevance and long-term engagement. In addition, ad systems often raise greater privacy concerns due to extensive user profiling [17], [18].

Table 2. Comparison of ad targeting and organic retrieval systems

Aspect	Ad Targeting Systems	Organic Retrieval Systems
Objective	Maximize advertising revenue and conversion rates	Improve user satisfaction and engagement
Primary data source	User demographics, browsing history, behavioral tracking	Historical interactions, preferences, and consumption patterns
Optimization target	Click-through rate (CTR), conversion rate, advertising ROI	Relevance, diversity, long-term user engagement
Evaluation metrics	Cost per click (CPC), CTR, conversion rate, revenue	DAU/MAU, retention rate, user engagement metrics
Business role	Direct monetization through advertising	Indirect revenue through user retention and platform growth
Privacy concerns	High due to extensive user profiling	Moderate, primarily focused on interaction data
Typical industrial examples	Google Ads, Facebook Ads, programmatic advertising	Netflix recommendations, YouTube recommendations, Spotify playlists

As shown in Table 2, although ad targeting and organic retrieval systems share similar underlying retrieval mechanisms, they differ significantly in their objectives, data usage, and evaluation metrics, leading to different optimization strategies in real-world systems. A key insight is that although ad targeting and organic retrieval systems share similar retrieval mechanisms, their optimization objectives lead to fundamentally different system designs. Ad systems prioritize short-term revenue metrics such as CTR and conversion rate, while organic systems emphasize long-term user engagement, diversity, and retention. This difference significantly influences how retrieval models are trained, tuned, and deployed in real-world systems.

5. METRICS AND EXPERIMENTATION

5.1. Metrics

Evaluation metrics and user experiments are widely used to assess recommendation system performance [21], [26]. In large-scale recommendation and advertising systems, commonly used metrics include click-through rate (CTR), area under the receiver operating characteristic curve (AUC), precision, and recall [26], [45]. CTR measures the proportion of impressions that result in user clicks and is widely used in advertising systems. Precision and recall evaluate the relevance of recommended items, while AUC reflects the model's ability to correctly rank positive interactions higher than negative ones. These metrics are typically used in offline evaluation before models are deployed to online systems [26], [45].

Content recommendation systems primarily focus on metrics that reflect user engagement and retention [21], [48], [49], such as monthly active users (MAU), daily active users (DAU), and user retention rates [21]. These systems aim to increase user engagement and retention through personalized content. The ultimate goal is to increase user loyalty and satisfaction, leading to sustained growth in user base and activity levels.

In addition to these high-level metrics, more granular metrics like clicks, likes, and follows are also crucial. These metrics are particularly sensitive to algorithmic updates and provide immediate feedback on user engagement with the recommended content. They are closely monitored to gauge the effectiveness of the recommendation algorithms and to make timely adjustments that can further enhance user interaction and satisfaction.

In contrast, ad systems operate within a three-party ecosystem involving advertisers, the platform, and users. Each party has distinct interests and metrics for success. Advertisers are primarily concerned with metrics such as cost per click (CPC) and conversion rates [17], [18], [45], [48], which indicate the effectiveness and efficiency of their ad spend. The platform, on the other hand, measures revenue generated from ad placements, aiming to maximize overall profitability. Users' interests lie in the relevance and non-intrusiveness of the ads they encounter; high relevance can enhance user experience, while irrelevant or excessive ads can detract from it. Balancing these three sets of interests is crucial for the success of ad systems.

5.2. Controlled experimentation framework

To evaluate and optimize both content recommendation and ad systems, controlled A/B experimentation frameworks [26], [34], [45] are commonly used. In this setup, users are randomly divided into control and treatment groups. The control group continues to experience the system as usual, while the treatment group is exposed to a new feature or change being tested. By comparing the behavior and outcomes of these two groups, the impact of the new feature can be accurately assessed. As illustrated in Figure 5, the A/B testing workflow involves splitting users into experimental groups, collecting behavioral data, and analyzing the results to guide system improvements [27].

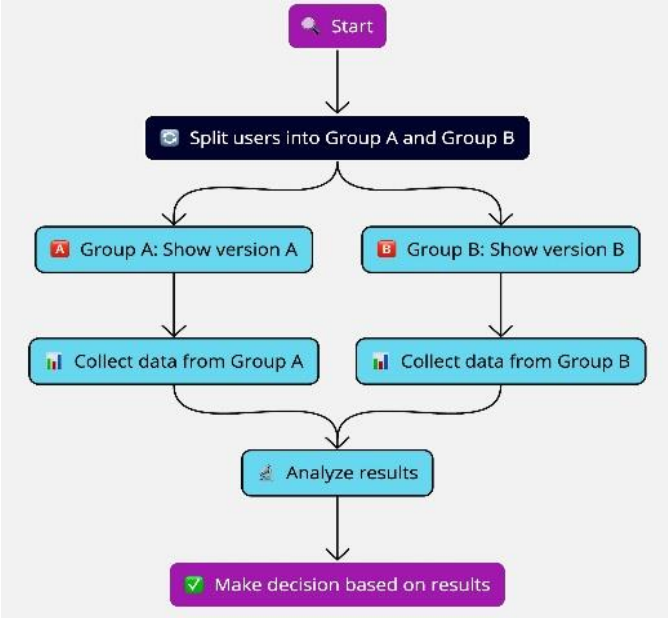

Figure 5. A/B experimentation workflow

However, A/B experiments in these systems come with unique challenges. For user experiments in content recommendation, there is a risk of traffic stealing, where the treatment group may inadvertently attract users who would otherwise belong to the control group. This can skew the results, making it difficult to measure the true effect of the experiment. Metrics such as clicks, likes, and follows are particularly sensitive to algorithmic updates and need to be closely monitored during these experiments to assess the immediate impact on user engagement [26].

When experimenting with content creators or advertisers, it's essential to separate users and creators to avoid "ghost experimentation" differences. This means ensuring that users and content creators (or advertisers) do not cross-contaminate the control and treatment groups. For instance, if an experiment is run with a subset of content creators, the audience for these creators should be evenly distributed between control and treatment groups to ensure accurate measurement of the experiment's impact.

For ad experiments, it is also critical to ensure that both the treatment and control groups have the same budget. This helps to maintain a fair comparison between the two groups and ensures that any differences in performance can be attributed to experimental changes rather than budget discrepancies.

6. CONCLUSION

Retrieval algorithms play a central role in modern advertising and content recommendation systems by enabling efficient matching between users and relevant items. This survey reviewed representative retrieval approaches used in both advertising recommendation and organic content recommendation systems, including traditional retrieval techniques and neural retrieval architectures such as the two-tower model. These methods form the foundation of large-scale recommendation pipelines by enabling efficient candidate generation.

Despite significant progress, several challenges remain in real-world systems, including data sparsity, cold start issues, data quality, and privacy concerns. Practical solutions such as transfer learning, synthetic data generation, and federated learning have been explored to mitigate these limitations. In addition, infrastructure scalability is a key consideration in industrial deployments, where techniques such as approximate nearest neighbor search, distributed retrieval pipelines, and hardware acceleration (*e.g.*, GPU/TPU-based systems) are widely adopted to support efficient large-scale retrieval.

Looking forward, retrieval algorithms are expected to continue evolving alongside advances in machine learning and large-scale computing. Emerging research directions include deeper integration between retrieval architectures and LLMs, as well as improved system designs that balance personalization, scalability, and privacy. Overall, retrieval-based architecture is expected to remain a fundamental component of modern recommendation systems. Recent advances in neural ranking and dense retrieval further highlight the rapid evolution of retrieval architectures, emphasizing the importance of scalable semantic matching and efficient representation learning in modern information systems. These trends suggest that future recommendation systems will increasingly rely on unified retrieval architectures that integrate semantic understanding, scalability, and real-time decision-making. This work provides a practical reference for both academic research and industrial deployment of retrieval-based recommendation systems.

FUNDING INFORMATION

This research received no external funding.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Yu Zhao	✓	✓												
Fang Liu					✓	✓			✓					
Yuan Yuan										✓				
Yifan Dang										✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38, 2020, doi: 10.1145/3285029.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005, doi: 10.1109/TKDE.2005.99.

[3] Y. Liu, Y. Wang, M. Zhang, S. Ma, and L. Ru, "A survey on neural ranking models for information retrieval," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, 2022.

[4] B. Mitra and N. Craswell, "Neural information retrieval: A literature review," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–38, 2018.

[5] J. Lin, X. Ma, S. Lin, and S. Yu, "Dense retrieval for information access: a survey," *ACM Transactions on Information Systems (TOIS)*, vol. 41, no. 4, 2023.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009, doi: 10.1109/MC.2009.263.

[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *26th International World Wide Web Conference, WWW 2017*, 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[8] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, vol. 33, pp. 9459–9474.

[9] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1877–1901.

[10] J. Achiam *et al.*, "GPT-4 technical report," *OpenAI*, *arXiv preprint arXiv:2303.08774*, 2023, [Online]. Available: https://arxiv.org/pdf/2303.08774.

[11] K. Zhou, "Large language models are powerful sequential recommenders," *ACM Conference on Recommender Systems (RecSys '23)*, 2023.

[12] J. Zhang, K. Bao, Y. Zhang, W. Wang, F. Feng, and X. He, "Large language models for recommendation: Progresses and future directions," *WWW 2024: Companion - Companion Proceedings of the ACM Web Conference*, pp. 1268–1271, 2024, doi: 10.1145/3589335.3641247.

[13] L. Wu *et al.*, "A survey on large language models for recommendation," *World Wide Web*, vol. 27, no. 5, 2024, doi: 10.1007/s11280-024-01291-2.

[14] F. Scholer, H. E. Williams, J. Yiannis, and J. Zobel, "Compression of inverted indexes for fast query evaluation," in *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 2002, pp. 222–229, doi: 10.1145/564376.564416.

[15] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009, doi: 10.1561/1500000019.

[16] B. Mitra and N. Craswell, "An introduction to neural information retrieval," *Foundations and Trends in Information Retrieval*, vol. 13, no. 1, pp. 1–129, 2018, doi: 10.1561/1500000061.

[17] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen, "How much can behavioral targeting help online advertising?," in *WWW'09 - Proceedings of the 18th International World Wide Web Conference*, 2009, pp. 261–270, doi: 10.1145/1526709.1526745.

[18] A. Goldfarb and C. Tucker, "Implications of 'online display advertising: Targeting and obtrusiveness,'" *Marketing Science*, vol. 30, no. 3, pp. 413–415, 2011, doi: 10.1287/mksc.1100.0634.

[19] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019, [Online]. Available: http://arxiv.org/abs/1907.11692.

[20] P. Resnick and H. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[21] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 191–198, doi: 10.1145/2959100.2959190.

[22] Q. Liu *et al.*, "Multimodal recommender systems: A survey," *ACM Computing Surveys*, vol. 57, no. 2, pp. 1–17, 2024, doi: 10.1145/3695461.

[23] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009*, 2009, pp. 452–461.

[24] D. Agarwal and M. Gurevich, "Fast top-k retrieval for model-based recommendation," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM)*, 2012, pp. 483–492.

[25] B. Mitra, F. Diaz, and N. Craswell, "Learning to match using local and distributed representations of text for web search," *26th International World Wide Web Conference, WWW 2017*, vol. 58, no. 2, pp. 1291–1299, 2017, doi: 10.1145/3038912.3052579.

[26] X. Yi *et al.*, "Sampling-bias-corrected neural modeling for large corpus item recommendations," in *RecSys 2019 - 13th ACM Conference on Recommender Systems*, 2019, pp. 269–277, doi: 10.1145/3298689.3346996.

[27] Q. Ai, K. Bi, J. Guo, and W. B. Croft, "Learning a deep listwise context model for ranking refinement," in *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, 2018, pp. 135–144, doi: 10.1145/3209978.3209985.

[28] Y. Zhang, Q. Ai, X. Chen, and W. B. Croft, "Joint representation learning for top-N recommendation with heterogeneous information sources," *International Conference on Information and Knowledge Management, Proceedings*, vol. Part F1318, no. 1, pp. 1449–1458, 2017, doi: 10.1145/3132847.3132892.

[29] X. Li, L. Sun, M. Ling, and Y. Peng, "A survey of graph neural network based recommendation in social networks," *Neurocomputing*, vol. 549, 2023, doi: 10.1016/j.neucom.2023.126441.

[30] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, "Neural graph collaborative filtering," *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, vol. 37, no. 4, pp. 165–174, 2019, doi: 10.1145/3331184.3331267.

[31] X. Wang, X. He, Y. Cao, M. Liu, and T. S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 950–958, doi: 10.1145/3292500.3330989.

[32] W. Fan *et al.*, "Graph neural networks for social recommendation," *arXiv:1902.07243*, pp. 1–11, 2019.

[33] X. He, K. Deng, X. Wang, Y. Li, Y. D. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648, doi: 10.1145/3397271.3401063.

[34] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: A survey," *ACM Computing Surveys*, vol. 55, no. 5, 2023, doi: 10.1145/3535101.
[35] C. Gao, X. He, Y. Li, and D. Jin, "Graph neural networks for recommender systems: Challenges and opportunities," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
[36] Z. Liang, H. Ding, and W. Fu, "A survey on graph neural networks for recommendation," *Proceedings - 2021 International Conference on Culture-Oriented Science and Technology, ICCST 2021*, pp. 383–386, 2021, doi: 10.1109/ICCST53801.2021.00086.
[37] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, 2019, pp. 3307–3313, doi: 10.1145/3308558.3313417.
[38] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 188–197, doi: 10.18653/v1/D19-1018.
[39] D. Li, H. Qu, and J. Wang, "A survey on knowledge graph-based recommender systems," *Proceedings - 2023 China Automation Congress, CAC 2023*, vol. 34, no. 8, pp. 2925–2930, 2023, doi: 10.1109/CAC59555.2023.10450693.
[40] Y. Deldjoo, T. Di Noia, and F. Merra, "Content-aware and multimodal recommender systems: State of the art and future challenges," *Information Fusion*, vol. 89, pp. 1–16, 2023.
[41] Z. Sun, X. Yu, J. Guo, X. Cheng, and J. Ren, "A survey on contrastive learning for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2023.
[42] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, "Self-supervised learning for recommender systems: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 1, pp. 335–355, 2024, doi: 10.1109/TKDE.2023.3282907.
[43] H. Zhao, Q. Yao, J. Li, Y. Song, and D. Yin, "Meta-learning based recommendation: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 6, pp. 1–38, 2023.
[44] C. Chronis, K. Berberidis, and G. Pallis, "Federated learning in privacy-preserving recommender systems: A survey," *IEEE Open Journal of the Computer Society*, vol. 5, 2024.
[45] H. B. Mcmahan *et al.*, "Ad click prediction: A view from the trenches," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *KDD'13*, Chicago, Illinois, USA, 2013, vol. Part F1288, pp. 1222–1230, doi: 10.1145/2487575.2488200.
[46] X. He and T. S. Chua, "Neural factorization machines for sparse predictive analytics," in *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 355–364, doi: 10.1145/3077136.3080777.
[47] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in *RecSys 2020 - 14th ACM Conference on Recommender Systems*, 2020, pp. 240–248, doi: 10.1145/3383313.3412488.
[48] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Computing Surveys*, vol. 54, no. 7, pp. 1–38, Sep. 2022, doi: 10.1145/3465401.
[49] B. zs Hidasi and A. Karatzoglou, "Recurrent neural networks with Top-k gains for session-based recommendations," *International Conference on Information and Knowledge Management, Proceedings*, vol. 36, no. 4, pp. 843–852, 2018, doi: 10.1145/3269206.3271761.

BIOGRAPHIES OF AUTHORS

Yu Zhao is a machine learning engineer specializing in large-scale industrial artificial intelligence and recommendation systems. He currently works as a senior machine learning engineer, focusing on modeling, optimization, and data-driven system design. He has previously held machine learning and AI roles at major technology companies including Meta and Twitter, where he contributed to the development of scalable recommendation and advertising systems. His work focuses on user representation learning, large-scale ranking systems, and production-level machine learning infrastructure. His research interests include information retrieval, recommendation systems, and large language models, particularly in the context of industrial retrieval pipelines. He has a multidisciplinary background in machine learning and business analytics and holds an MBA from the University of Toronto. He can be contacted at email: yzqr.zhao@rotman.utoronto.ca.

Fang Liu received the Master of Business analytics degree from Saint Peter's University and the Master of Advanced Management degree from Yale School of Management. She also holds an MBA from Fudan University School of Management. She has more than ten years of professional experience in consulting, financial services, and technology-driven business transformation. She previously served as senior manager of lending business management at American Express in New York, where she worked on AI-driven credit modeling and data analytics for small-business lending. Earlier in her career, she worked at Deloitte on enterprise digital transformation and large-scale information systems implementation projects. Her research interests include applied artificial intelligence, machine learning, and data-driven decision systems in real-world industrial settings. She focuses on bridging artificial intelligence methodologies with real-world decision-making systems and industrial applications. She can be contacted at email: fangliu435@gmail.com.

Yuan Yuan received the B.E. degree in computer science and is currently pursuing the M.S. degree in computer science at Stevens Institute of Technology. He has several years of professional experience in software engineering and enterprise information systems development. Prior to his graduate studies, he participated in enterprise digital transformation and large-scale software implementation projects, including work related to enterprise system integration and cloud-based technology solutions at Deloitte. He has also contributed to the development and deployment of enterprise software systems and cloud infrastructure platforms supporting large-scale business applications. His research interests include machine learning, recommendation systems, information retrieval, distributed systems, and cloud computing platforms. His work focuses on scalable machine learning systems and data-driven architectures for intelligent information systems, recommendation technologies, and large-scale data processing. He can be contacted at: yuan903@gmail.com.

Yifan Dang received the Master of Science degree in quantitative finance from the Robert H. Smith School of Business, University of Maryland, College Park, MD, USA. He has a background in data analytics and quantitative modeling, with a focus on applying machine learning techniques to real-world financial and business problems. His work includes data-driven modeling, financial risk analysis, and the application of artificial intelligence in business decision systems. He has also contributed to research on retrieval algorithms and recommendation systems, particularly in the context of large-scale industrial data pipelines. His research interests include machine learning, information retrieval, recommendation systems, and AI-driven analytics in financial services. He can be contacted at email: yifandang@gmail.com.