Answering Narrative-Driven Recommendation Queries via a Retrieve–Rank Paradigm and the OCG-Agent

Yunxiao Shi[1,*], Haoning Shang[2], Xing Zi[1], Wujiang Xu[3], Yue Feng[4], Min Xu[1,†],

[1]The University of Technology Sydney, [2]The University of New South Wales, [3]Rutgers University, [4]The University of Birmingham

[*]First author. [†]Corresponding author. Email: min.xu@uts.edu.au

Abstract

Narrative-driven recommendation queries are common in question-answering platforms, AI search engines, social forums, and some domain-specific vertical applications. Users typically submit free-form text requests for recommendations, e.g., "Any mind-bending thrillers like Shutter Island you'd recommend?" Such special queries have traditionally been addressed as generic QA task under the RAG paradigm. This work formally introduces narrative recommendation as a distinct task and contends that the RAG paradigm is inherently ill-suited for it, owing to information loss in LLMs when retrieving information from from multiple long and fragmented contexts, and limitations in ranking effectiveness. To overcome these limitations, we propose a novel retrieve-rank paradigm by theoretically demonstrating its superiority over RAG paradigm. Central to this new paradigm, we specially focus on the information retrieval stage and introduce **O**pen-domain **C**andidate **G**eneration (OCG)-Agent that generatively retrieves structurally adaptive and semantically aligned candidates, ensuring both extensive candidate coverage and high-quality information. We validate effectiveness of new paradigm and OCG-Agent's retrieve mechanism under real-world datasets from Reddit and corporate education-consulting scenarios. Further extensive ablation studies confirming the rationality of each OCG-Agent component. The code is available at [1].

1 Introduction

The narrative-driven recommendation (NDR) (Bogers and Koolen, 2018; Eberhard et al., 2019; Afzali et al., 2021)—which leverages users' explicitly stated narrative queries to suggest personalized items—has recently garnered attention through the application of large language models (LLMs) (Eberhard et al., 2025; Mysore et al., 2023; Zhu et al., 2025b), because of their exceptional semantic understanding (Zhu et al., 2024), advanced reasoning, and zero-shot adaptability (Brown et al., 2020; OpenAI et al., 2024).

Beyond relying solely on the parameterized knowledge of LLMs for direct answer generation, augmenting LLMs with externally retrieved evidence through a Retrieval-Augmented Generation (RAG) framework has been shown to substantially enhance accuracy, credibility, and timeliness (Lewis et al., 2020; Karpukhin et al., 2020; Izacard et al., 2023; Gao et al.). Commercial AI search engines are prime examples of this paradigm in action, demonstrating strong feasibility for both question answering (Soto-Jiménez et al., 2024; Fernández-Pichel et al., 2025) and autonomous information retrieval (Amer and Elboghdadly, 2024; Jiang et al., 2025; Lu et al., 2025). However, the effectiveness of these systems in answering narrative recommendation queries remains largely unexamined. A substantial portion of real-world queries—from advice-seeking on social platforms (e.g., Reddit, REDnote) to domain-specific consultancy requests—naturally conform to a narrative-driven recommendation format. Consequently, a key open question is: *How do generic QA- and information-search-oriented AI search engines perform when tasked with narrative-driven recommendation queries?*

To investigate this, we conducted exploratory experiments to evaluate the performance of several AI search engines on narrative-driven movie recommendation queries (§3). Surprisingly, these systems consistently underperformed standalone LLMs, underscoring the limited efficacy of the RAG paradigm. Our diagnostic analysis revealed two critical limitations responsible for this gap: **Low candidate recall constrains the recommendation performance ceiling**, and **Insufficient candidate information impedes accurate ranking**. These findings indicate that resolving these retrieval bottlenecks is essential. In particular, adopt-

[1]https://github.com/Ancientshi/OCG-Agent

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 13193–13214

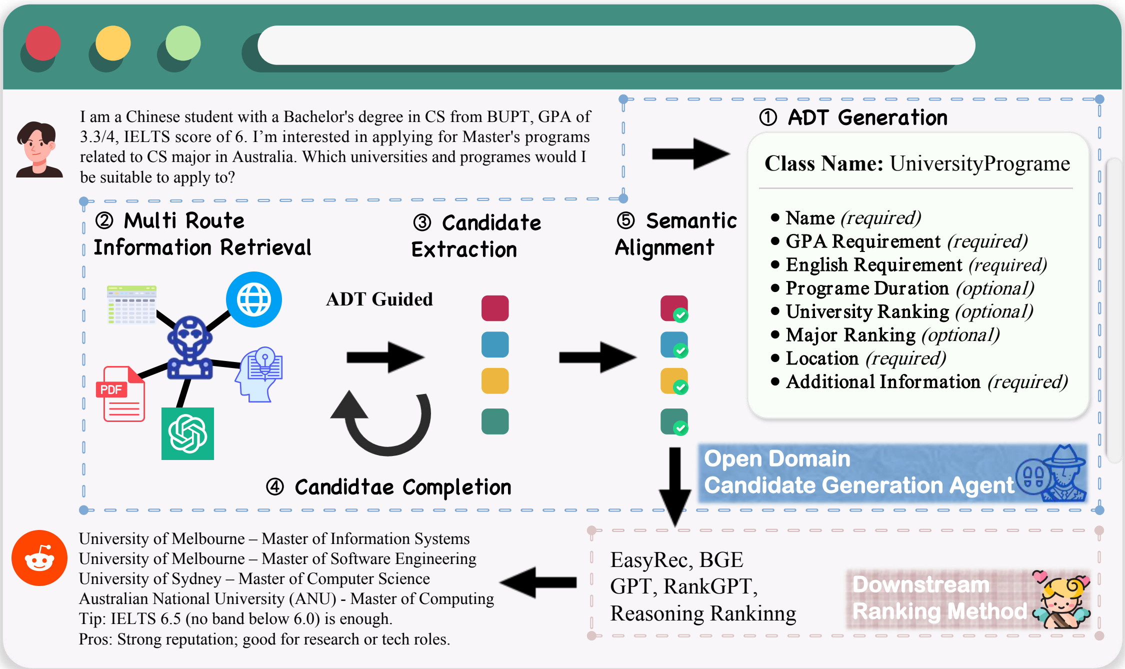

Figure 1: OCG-Agent in Narrative-driven Recommendation Task.

ing a wide-and-deep information retrieval strategy offers a promising avenue for overcoming these challenges and elevating the performance ceiling of narrative recommendation tasks.

Additionally, we realize there exist information-loss issue when applying the RAG paradigm to narrative recommendation. Factual snippets retrieved from heterogeneous sources are fragmented across numerous documents and cluttered with noise; these long, interleaved contexts overwhelm LLMs, blur entity boundaries, and degrade extraction precision (Jin et al., 2025a; Liu et al., 2023c). Furthermore, LLMs exhibit inherent shortcomings in rank list-generation tasks—such as position and popularity bias (Hou et al., 2024), a mismatch between token-prediction objectives and listwise ranking goals (Sun et al.), and accuracy degradation when handling large-scale candidate paragraphs (Liu et al., 2023b). **The RAG paradigm is inherently ill-suited to narrative recommendation, as it incurs information-extraction losses and combined with suboptimal ranking capability.** We develop a rigorous theoretical analysis that demonstrates how these two intertwined deficiencies critically erode recommendation quality (§6).

Motivated by these findings, we makes the following contributions in this work. ❶ ***Formalization of the narrative recommendation task and introduction of a retrieval–ranking paradigm beyond retrieve-then-read (§ 2).*** We also provide a theoretical guarantee that our novel paradigm will firmly deliver outperform than RAG paradigm (§ 6). ❷ ***We introduce OCG-Agent, a novel open-domain information-retrieval agent (§ 5),*** which specifically designed to enable wide and in-depth candidate retrieval for narrative recommendation queries. ❸ ***We verify the effectiveness of OCG-Agent's wide-deep retrieval mechanism and the new retrieve-rank paradigm on Real-World Reddit and Corporate Datasets.*** Both RAG and retrieve–rank implementations consistently outperform LLM-based strong baselines, unlike current AI search engines, and advanced deep-research products lagging behind. Besides, our retrieve–rank paradigm achieves a 18.5%, and 27.3% improvement in NDCG on the movie dataset and education dataset, respectively, compared to conventional retrieve–then–read variants. ❹ ***Critical Findings in Ablation Study (§ 9).*** The ablation on OCG-Agent demonstrates that expanding retrieval coverage improves overall performance but can also induce retrieval saturation, leading to ranking degradation. Moreover, employing LLM-based generative retrieval is particularly effective for hard-to-retrieve queries. By progressively deepening the retrieval process and enriching each candidate's information, the pipeline's recommendation accuracy is incrementally enhanced—especially in niche, domain-specific contexts where LLMs' parameterized knowledge is insufficient. Finally, semantic alignment further boosts precision in detail-sensitive domains, e.g, education.

2 Preliminary

2.1 Narrative-driven Recommendation Task

Definition 1. Let q represent user query, and $I^*(q)$ denotes the ground truth recommended items. $\mathcal{Q}$ denote the space of user queries. $\mathcal{I}$ denote the space of candidate items. A Top-K narrative recommender is a function

$$F : \mathcal{Q} \longrightarrow I^K, q \longmapsto F(q) = [\ell_1, \ldots, \ell_K],$$

where each $\ell_j \in I$ is a textual identifier (e.g., a movie title) and the list is ordered by descending relevance, $\mathrm{rel}(\ell_1, q) \geq \mathrm{rel}(\ell_2, q) \geq \ldots \geq \mathrm{rel}(\ell_K, q)$. Here $\mathrm{rel}(\cdot)$ is an implicit scoring function. Any candidate retrieval or re-rank procedure is encapsulated inside F.

2.2 RAG Paradigm for Question Answering

Given a narrative query $q \in \mathcal{Q}$, the system retrieves a knowledge set

$$\mathcal{E}(q) = \{ d_1, \ldots, d_M \}, \quad M = |\mathcal{E}(q)|,$$

where d_i denotes retrieved document may contain candidates' information (e.g., title, description.) The combined input $(q, \mathcal{E}(q))$ is then fed into a generative LLM f_θ, parameterized by θ, which directly produces a ranked list of items:

$$\widehat{F}(q) = f_\theta(q, \mathcal{E}(q)) = [\ell_1, \ldots, \ell_K].$$

3 Motivational Experiments

Experiment Setup. We evaluated 30 benchmark movie-recommendation queries (Eberhard et al., 2019, 2024, 2025) by submitting each to three commercial AI search engines (ChatGPT-Search, Perplexity-Sonar and Gemini-Search) and to GPT-4o-mini. All systems used the same prompt and their raw outputs were normalized into JSON-formatted ranked lists. We then computed Precision@10, Recall@10 and NDCG@10 for each top-10 list. Further details on the prompt template, post-processing and evaluation protocol are provided in Appendix A.

Result Analysis. Figure 2 presents the mean performance across all queries. We can observe that **across all three ranking metrics, AI search engines underperform GPT-4o-mini by an average relative drop of over 20%.** We hypothesize two primary factors underlying this gap. First, **limited candidate recall** in the search engines imposes a ceiling on achievable ranking performance. Second, **insufficient information richness** impairs the LLM's ability for accurately generating ranked recommendations.

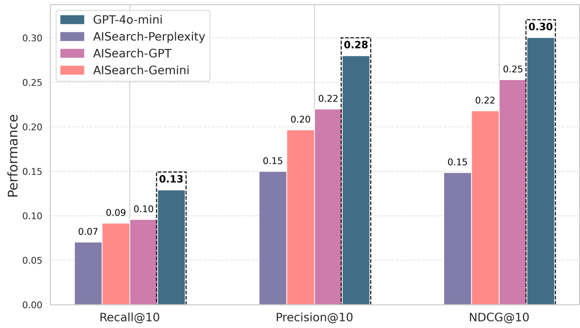

Figure 2: Performance Comparison Between Large Language Models and AI Search Engines for Narrative Recommendation at Top@10.

Table 1: Deficiency Investigation Results

Setting	Precision@10	Recall@10	NDCG@10
A	0.2266	0.0989	0.2485
B	0.5133	0.2276	0.5366
C	0.2433	0.1091	0.2725

Deficiency Exploration. To validate our hypotheses, we designed three experimental setting. **Setting A** serves as our baseline retrieve-then-read (RAG) pipeline: we query the Serper API[2] for web search results, extract content with Docling (Team, 2024), and supply only the retrieved movie titles as external knowledge to the prompt. In **Setting B**, we include all ground-truth movie titles, then add negative candidates (non-relevant movies) to maintain a realistic positive-to-negative ratio of approximately 1:3. This ratio mimics the real-world web retrieval system where relevant results are naturally sparse. By controlling the candidate pool composition in this way, we can isolate the impact of retrieval coverage on the ranking performance, independent of other retrieval artifacts. Finally, **Setting C** enriches the prompt's external knowledge with both movie titles and their associated metadata, allowing us to evaluate the benefit of richer contextual information. These comparisons disentangle the effects of candidate-set size versus information richness on recommendation performance. Table 1 reports Precision@10, Recall@10, and NDCG@10 for each setting. Setting B yields a dramatic improvement over the baseline—confirming that retrieval coverage is critical—while Setting C produces a modest gain, underscoring the value of enriched metadata.

[2] https://serper.dev

4 Retrieve–Rank Paradigm for NDR

We advocate the classical two-stage *retrieve–rank* paradigm, long established and effective in traditional recommender systems.

Define $C(q) = Retrieve(q) \subseteq \mathcal{I}, |C(q)| = N \gg K$, where Retriever employs broad retrieval strategies to assemble a large, high-coverage candidate set. Define a $Reranker$ takes the narrative query q together with its candidate list $C(q) = \{c_1, \ldots, c_N\}$ and directly returns an ordered Top-K prediction $\widehat{F}(q) = Rerank\big(q, C(q)\big), |\widehat{F}(q)| = K$.

This retrieve–rank paradigm provides the conceptual footing for our Open-Domain Candidate Generation Agent (OCG-Agent; see § 5), which instantiates the $Retrieve(\cdot)$ stage. The subsequent $Rerank(\cdot)$ module is deliberately modular: such as point-wise re-ranker (Cheng et al., 2022; Chen et al., 2024), LLM-driven re-ranker (Jin et al., 2025c), RAG-based re-ranker (Zhao et al., 2024c), or agentic ranking techniques (Jin et al., 2025b; Sun et al.) based on LLM's reasoning and planing ability (Jin et al., 2024b,a; Shi et al., 2025b).

5 Open-domain Candidate Generation

5.1 ADT Generation

We map each narrative query $q \in \mathcal{Q}$ into a structured abstract data type (ADT) $t \in \mathcal{T}$ for representing a candidate:

$$t \;=\; \big\{(a_j,\, v_j,\, \mathbb{I}_j)\big\}_{j=1}^{m},$$

where a_j is the attribute name, v_j is its instantiated value (possibly empty), and $\mathbb{I}_j \in \{\mathsf{REQUIRED}, \mathsf{OPTIONAL}\}$ indicates whether a_j is essential. To guarantee a minimal schema, we include two mandatory fields *Name* and *AdditionalInformation*, where *Name* uniquely identifies the candidate and *AdditionalInformation* holds extensible auxiliary metadata. We define $f_\theta^{ADT} : \mathcal{Q} \longrightarrow \mathcal{T}$ and implement this mapping via chain-of-thought prompting:

$$t \sim f_\theta\big(t \mid \text{prompt}_{ADT}(q)\big) = f_\theta^{ADT}(q). \quad (1)$$

This formulation treats each REQUIRED attribute as a direct filter drawn from the query—e.g., program start semester, GPA threshold for educational recommendations—it achieves precise alignment with user needs. Besides, the fixed schema imposes a uniform structure that supports fair comparisons across heterogeneous web sources. Moreover, whenever a required field is missing ($v_j = \emptyset$), the system automatically invokes the reflect-and-augment routine (see § 5.4), guaranteeing iterative completion of all critical attributes.

5.2 Multi-Route Information Retrieval

OCG-Agent pursues a large, high-coverage candidate set $C(q)$ through an agentic multi-route retrieval routine: by chaining autonomous function calls, it composes and executes complementary retrieval routes that sweep heterogeneous data sources in parallel—a strategy long applied in practical recommender systems (Huang et al., 2024; Nie et al., 2022; Huang et al., 2025). This process can be formally described as:

$$\mathcal{P}(q) = \big\{(r_i,\, k_i)\big\}_{i=1}^{n} \;\sim\; f_\theta^{\text{Rewrite}}(q), \quad (2)$$

where each r_i is a callable retrieval function and k_i are its subquery parameters. We employ four complementary retrieval channels. The *Web search* route, denoted as $r_{\text{web}}(k)$, leverages Docling (Team, 2024) for webpage content extraction. For retrieving knowledge from a specific domain, we use *Vector search* $r_{\text{vector}}(k)$ implemented by Chroma+LangChain for retrieving most relevant documents according to semantic similarity. We also use *Structured query* route denoted by $r_{\text{SQL}}(k)$, via MindSQL for relational data lookup. And finally completed with *Generative LLM* $r_{\text{LLM}}(k)$, for directly generating information based on LLMs' parameterized knowledge, which is effective for retriving useful information that is hard to be retrieved from internet. The union of the retrieved knowledge fragments forms aggregated knowledge base:

$$\mathcal{E}(q) = \bigcup_{i=1}^{n} r_i(k_i) = \{\, d_1, \ldots, d_M\}, \quad (3)$$

Here, each retrieval route contributes a subset of knowledge fragments to the collective repository.

5.3 Candidate Extraction

We deploy a parallel fragment-level LLM candidate extract followed by aggregation that deduplicates and unifies the candidate set. We extract candidates in parallel:

$$\mathcal{C}^{(j)}(q) = [c_1^{(j)}, \ldots, c_{n(j)}^{(j)}] \sim f_\theta^{\text{Extract}}(d_j,\, t), \quad (4)$$

where t is the Abstract Data Template (ADT). Any ADT field unsupported by d_j is marked `NOT FOUND`.

We then aggregate all local sets into a unified candidate pool: $\mathcal{C}(q) = \bigcup_{j=1}^{n} \mathcal{C}^{(j)}(q)$, For candidates c appearing in multiple $\mathcal{C}^{(j)}(q)$, we consolidate their attributes via $c = \bigoplus_{j:\, c \in \mathcal{C}^{(j)}} c^{(j)}$, where

$\oplus$ merges complementary fields to yield enriched, consistent representations.

5.4 Reflective Completion for Attributes

Multi-route recall (§ 5.3) maximizes coverage, yet it often yields candidates with missing REQUIRED fields—attribute sparsity that hurts ranking accuracy (§ 3). OCG-Agent remedies this through a *reflect-and-complete* phase that audits each candidate and fills every mandatory attribute.

Problem Formulation. Represent a candidate as an attribute map $c = \{(a_j, v_j)\}_{j=1}^{m}, v_j \in \mathcal{V} \cup \{\varnothing\}$, where $\mathcal{V}$ is the space of admissible values. Let $\mathcal{A}_{req} \subseteq \{a_1, \dots, a_k\}$ denote the set of required attributes defined by the ADT schema. The *completion set* of c is $\mathcal{M}(c) = \{ a_j \in \mathcal{A}_{req} \mid v_j = \varnothing\}$. Our objective is to construct an operator

$$\mathcal{C} : c \longmapsto \hat{c}, \qquad \text{s.t.} \quad \mathcal{M}(\hat{c}) = \varnothing,$$

while preserving all previously verified values.

Targeted Deep Retrieval. For every missing attribute $a \in \mathcal{M}(c)$ we craft a query $k(c, a) = \mathrm{Compose}(c, a)$, which encodes both the candidate identifier (e.g., a movie title) and the attribute to be filled. Leveraging the multi-route retrieval module (§ 5.2), the OCG-Agent autonomously invokes several specialized retrievers—each probing a distinct search direction—and aggregates their outputs into the knowledge set $\mathcal{E}(c, a)$, which is then used to complete the required attribute.

Completion. We define a chain-of-thought prompt driven completion process as

$$v = f_\theta^{\text{COMP}}\big(k(c, a), \mathcal{E}(c, a)\big),$$

The candidate is updated in place,

$$\hat{c} = c \cup \big\{(a, v)\big\},$$

and the procedure iterates until $\mathcal{M}(\hat{c}) = \varnothing$. By integrating the explicit $\mathcal{M}(c)$ checklist with adaptive, attribute-targeted retrieval, the reflect-and-complete stage eliminates the extra LLM-mediated reflection step that conventional/deep research pattern RAG pipelines typically require.

5.5 Expert-Guided Semantic Normalisation

Even after attribute completion, values may remain *semantically incommensurable*. A canonical example is grade-point averages: Australia scales GPA on $0-7$, whereas the UK adopts $0-4$. Such incongruities bias similarity metrics and, in turn, downstream ranking.

Alignment operator. Let $c = \{(a_j, v_j)\}_{j=1}^{m}$ be a completed candidate and $\mathcal{A}_{sense}$ is the subset of *semantically sensitive* attributes. For every $a_j \in \mathcal{A}_{sense}$ we prompt LLMs with human expert-level domain knowledge $\mathcal{E}_{expert}$ (e.g. conversion formulae, ontologies, or policy tables) and apply

$$\bar{v}_j = f_\phi^{\text{Normalize}}\big(v_j, \mathcal{E}_{expert}\big),$$

which is implemented by an LLM prompted with chain-of-thought exemplars. Empirically, this step is often important in cross domain recommendations such as the cross-national education benchmark (§ 9.2), underscoring the necessity of expert-guided normalisation.

6 Theoretical Effect

Assumption 1. *The number of mentioned candidate in $\mathcal{E}(q)$ is N. There exists $\gamma \in [0, 1]$ such that only γN of the retrieved items survive inherent information loss of LLM in handling long context. OCG-Agent can often achieve $\lambda \to 1$, such that OCG-Agent successfully recognizes and extracts every item in the $\mathcal{E}(q)$ yielding $\lim_{\lambda \to 1} C(q) = N$. There exists $\rho \in [0, 1]$ satisfying $\Pr\big(\mathcal{E}(q) \supseteq \mathcal{I}^*_{\text{top}}(q)\big) \geq \rho$, where $\mathcal{I}^*_{\text{top}}(q) \subseteq \mathcal{I}$ is the true top-K relevant set. Leverage LLM for top-K ranking task yielding a accuracy of $\beta \in [0, 1]$. Moreover, a sophisticated re-ranker achieves an accuracy of $\alpha \in [0, 1]$ with $\alpha \geq \beta$.*

Theorem 1. *Under Assumption 1, the precision and recall of Retrieve-Read RAG paradigm and Retrieve-Rank satisfies*

$$\mathbb{E}[\text{P@}K_{\text{RAG}}] \geq \frac{\gamma N \rho \beta}{K}, \quad \mathbb{E}[\text{R@}K_{\text{RAG}}] \geq \frac{\gamma N \rho \beta}{|I^*(q)|},$$

$$\mathbb{E}[\text{P@}K_{\text{RR}}] \geq \frac{N \rho \alpha}{K}, \qquad \mathbb{E}[\text{R@}K_{\text{RR}}] \geq \frac{N \rho \alpha}{|I^*(q)|}.$$

It is apparent that this theoretical guarantee the proposed retrieve–rank paradigm delivers outperform RAG paradigm in both precision and recall. Detailed prove is provided in Appendix D.

7 Experiment

7.1 Experimental Setup

Datasets. We conduct experiments on two datasets chosen to mirror the *coverage* and *semantic-richness* deficiencies diagnosed in § 3. First, we adopt the REDDIT MOVIESUGGESTIONS benchmark originally released by Eberhard et al. (2019) and later reused by Eberhard et al. (2025). Second, we introduce an AUSEDU-NARRATIVES

Table 2: Comparison of metrics for different methods

Method	Movie			Education		
	Precision@10	**Recall@10**	**NDCG@10**	**Precision@5**	**Recall@5**	**NDCG@5**
GPT4o-mini	0.2800	0.1291	0.3003	0.2545	0.1682	0.3236
GPT4o	0.3133	0.1506	0.3451	0.3318	0.2617	0.3829
DeepSeek-R1	0.2767	0.1254	0.3068	0.3463	0.2712	0.3939
AISearch-Perplexity	0.1500	0.0705	0.1486	0.3473	0.2349	0.4519
AISearch-GPT	0.2200	0.0959	0.2531	0.3272	0.2227	0.4207
AISearch-Gemini	0.1967	0.0918	0.2180	0.3090	0.2041	0.4074
Open Deep Research	0.0931	0.0384	0.0804	0.2545	0.1650	0.3109
Perplexity Deep Research	0.2033	0.0876	0.2246	0.3090	0.2015	0.3423
Retrieve-then-Read	0.3143	0.1520	0.3324	0.3739	0.3073	0.5216
OCG-RankGPT	**0.3567**	**0.1832**	**0.3940**	**0.5342**	**0.4323**	**0.6641**

corpus consisting of 30 anonymised, real-world study-abroad counselling cases supplied by a local consultancy. A full description and ethical safeguards appears in Appendix B.

Evaluation. We measure Precision@k, Recall@k, and NDCG@k(Järvelin and Kekäläinen, 2002),(k=10 for movies, 5 for education). For each query, the OCG-Agent retriever runs once, after which RankGPT (Sun et al.) plays as re-ranker run for three times and we report the averaged value.

Baselines. We employ EasyRec (Cheng et al., 2022) as a first-stage re-ranking module to retain the top-50 candidates by pairwise score, and then apply RankGPT (Sun et al.), powered by O3-mini, for the final ranking. We denote our whole end-to-end pipeline method as OCG-RankGPT. We also have a baseline variant implemented by retrieve-read paradigm under the same external knowledge usage for fair comparison. Additionally, we further compare with the following categories of baselines. *LLM Direct*: GPT4o-mini, GPT4o and DeepSeek-R1 (DeepSeek-AI, 2025). *AI Search Engines*: Perplexity, GPT-AI Search, and Gemini Search. *Deep Research* (Lee, 2025), Perplexity [3] and Open Deep Research [4]. Further introduction for baselines are in Appendix C

We exclude traditional recommendation models (matrix factorization, TF-IDF, Doc2Vec) as baselines since prior work has shown they perform poorly on narrative-based queries and are inconsistent with human recommendations (Eberhard et al., 2019). We also exclude prompt engineering-based recommendation methods such as (Liu et al., 2023b) because prior work (Eberhard et al., 2025) shows that prompt engineering variants (zero-shot, few-shot) provide no improvement over basic prompting.

8 Results and Analysis

LLMs as Narrative Recommender is Effective in Generic Domains. As demonstrated in Table 2, LLM-based narrative recommenders consistently outperform both AI search engines and advanced deep research methods in the movie recommendation task. Notably, GPT-4o achieves performance closely approaching OCG-Agent, trailing by only approximately 4%. This highlights the inherent effectiveness and efficiency of harnessing the parameterized knowledge embedded in LLMs for general-domain narrative recommendation.

Wide-Deep Retrieval Enhanced RAG Paradigm Yields Performance Gains While Commercial Products Lag Behind. By treating narrative recommendation as a QA task under a RAG framework, our Retrieve–then–Read variant, enhanced by OCG-Agent's retrieved external knowledge, consistently outperforms standalone LLM approaches, commercial AI search engines, and advanced deep-research approaches. In the education domain, it achieves relative improvements of 7.97% in Precision@5, 13.31% in Recall@5, and 32.42% in NDCG@5 compared to DeepSeek-R1. This improvement validates the effectiveness of our wide-and-deep information retrieval efforts. In contrast, off-the-shelf commercial products not only exhibit degraded performance on the movie dataset but also deliver only marginal gains in educational recommendations, highlighting the limitations of their vanilla retrieval pipelines and underscoring the critical need for candidate-centric retrieval enhancements.

OCG-RankGPT Secures Marked Gains over RAG Paradigm. Compared to the re-

[3]https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research

[4]https://github.com/langchain-ai/open_deep_research

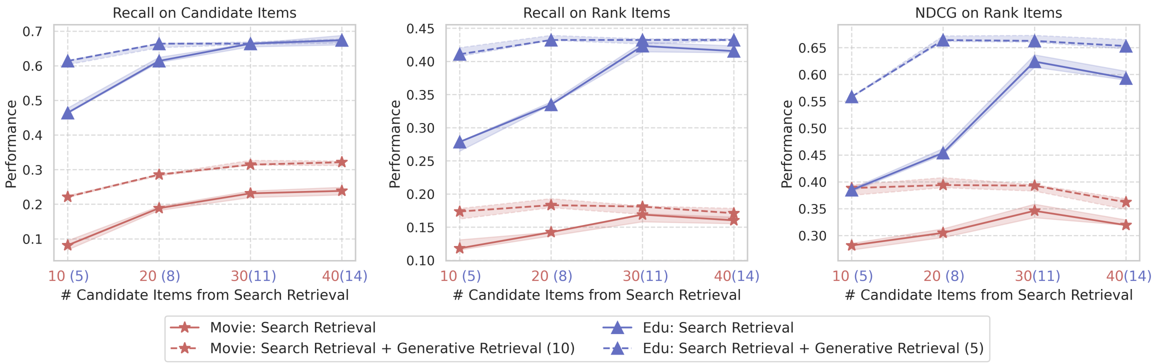

Figure 3: Impact of Increasing Candidate Items Generated by OCG-Agent on Recall of Candidate Generation, Recall of Reranking Results, and NDCG.

trieve–then–read variants, our OCG-RankGPT implementation delivers a 18.5% increase in NDCG@10 on the movie dataset and 27.3% uplift in NDCG@5 on education dataset. Since both approaches operate with the same amount of external knowledge, these improvements underscore the superiority of our novel retrieve–rank paradigm. Moreover, our theoretical performance bounds presented in Appendix D for the retrieve–rank framework exhibit strong concordance with the observed empirical gains.

9 Ablation Study

9.1 Impact of Strengthen Retrieval Breadth

We design our experiments to evaluate the impact of increasing retrieval channels on both candidate recall effectiveness and final recommendation performance. For the movie recommendation scenario, we employ four $r_{\text{web}}(k)$, and we select the top 10 candidates for each retriever. Additionally, we supplement this with one $r_{\text{LLM}}(k)$, which generates 10 candidate items. In the educational recommendation context, the retrieval strategy is tailored to domain-specific needs. We deploy just one $r_{\text{web}}(k)$ yielding the top 5 candidates, complemented by three $r_{\text{vector}}(k)$ with each one yields 3 candidates. We further integrate one $r_{\text{LLM}}(k)$, providing an additional 5 candidates. Performance was evaluated by measuring recall on candidates, recall and NDCG on the final ranks. The experimental results are visualized in Figure 3.

Retrieval Saturation and Ranking Degradation. Increasing the number of retrieval channels initially leads to substantial gains in candidate recall. However, these gains taper off as additional channels begin to yield overlapping or lower-quality items. This diminishing-return effect not only saturates recall improvements but also introduces redundancy and noise into the candidate pool. As a result, the final ranked performance—measured by recall and NDCG—can plateau or even degrade. These findings underscore a critical insight: beyond a certain point, expanding retrieval breadth harms rather than helps. Effective retrieval should therefore emphasize quality-aware selection over indiscriminate expansion to preserve downstream ranking fidelity.

Effectiveness of Generative Retrieval. Generative retrieval significantly boosts performance, particularly when conventional web search yields sparse results. In the movie domain, it reliably surfaced high-quality candidates that were difficult to obtain even with extensive web querying. In contrast, for the education domain, aggressive specific domain-based retrieval eventually caught up—but only with sustained effort. These results highlight the strategic value of generative retrieval: by leveraging the broad world knowledge encoded in large language models, it excels in both open-domain scenarios with hard-to-retrieve items and specialized domains requiring domain-specific expertise.

9.2 Impact of Strengthen Retrieval Depth

To isolate the effect of attribute completeness on ranking, we vary the fraction of required fields that are populated. Completeness is quantified as the ratio of filled essential attributes, and Table 3 summarizes the three tiers evaluated in our study.

Attribute Completeness Drives Ranking Precision. Augmenting each candidate with its full set of *required* attributes consistently raises ranking accuracy, but the scale of this benefit is decisively domain-specific. In the movie corpus, where LLMs already parameterized extensive cinematic knowledge, filling yields only modest gains. By con-

Table 3: Effect of Varying Levels of Information Completeness and Quality of Candidate Attributes on Ranking Performance

Dataset	Attribute Completeness	Required Attribute %	Ranking NDCG@10
Movies	Movie name only	0%	0.3241
	+ ADT Information (§5.1)	68%	0.3684
	+ Required Attribute Completion (§5.4)	87%	**0.3940**
	+ Sematic Alignment (§5.5)	87% (**0% updated**)	**0.3940**
Education	Program and university names only	0%	0.3978
	+ ADT Information (§5.1)	57%	0.4513
	+ Required Attribute Completion (§5.4)	92%	0.5849
	+ Sematic Alignment (§5.5)	92% (**27% updated**)	**0.6641**

trast, the education corpus shows a sharp accuracy jump once critical attributes—such as programme start term, GPA thresholds, and language requirements—are completed. These results reaffirm our central claim and highlight the strategic role of the attribute-completion module introduced in § 5.4: by invoking targeted deep retrieval to populate essential fields, it fortifies the retrieve–rank paradigm and delivers the fine-grained metadata indispensable for high-fidelity recommendations in domains where detail governs decision-making.

Semantic Alignment Lifts Precision in Detail-Sensitive Domains. Table 3 shows that normalize attributes boosts NDCG@10 on the education dataset from 0.585 to 0.664. Roughly 27% of its fields—chiefly exam scores and GPA formats—required conversion to a common schema. In contrast, the movies dataset have no change: its metadata are already standardised, so alignment touched 0% of attributes and left accuracy flat.

10 Related Work

10.1 Narrative-driven Recommendation

In a narrative-driven recommendation scenario (Bogers and Koolen, 2017), users articulate their needs in free-form prose—*"I'm looking for a mind-bending thriller like* Shutter Island*"*—and expect the system to return a ranked list of suitable items. Early methods grounded in classical text retrieval or embedding matching (Eberhard et al., 2024, 2020, 2019) struggled to capture the subtle intent encoded in such narratives and therefore achieved only modest accuracy. The advent of LLMs has transformed this landscape. Recent studies have demonstrated LLMs' potential for a wide range of recommendation tasks (Zhu et al., 2025a; Lubos et al., 2024; Dai et al., 2023; He et al., 2023; Liu et al., 2024; Feng et al., 2023; Hao et al., 2025; Chen et al., 2025; Shi et al., 2025a; Xu et al., 2025). Notably, Eberhard et al. (2025) report that GPT-class models surpass strong embedding baselines such as doc2vec (Le and Mikolov, 2014) on Reddit movie-suggestions.

In this study, we extend the narrative-driven recommendation beyond forum scenarios (Eberhard et al., 2025) to diverse real-world contexts—AI search engines, agentic consulting, question-answering systems, and social media posts—where users express recommendation requests as free-form narratives. Furthermore, we formally define this task and advocate a new retrieve-rank paradigm as solution beyond RAG.

10.2 Information Retrieval in LLM Era

Retrieval-Augmented Generation (RAG) has evolved from the foundational retrieve–then–read pipeline (Lewis et al., 2020; Karpukhin et al., 2020; Izacard et al., 2023) to modular architectures integrating advanced plug-in components (e.g., (Gao et al., 2024; Shi et al., 2024; **?**; Ma et al., 2023; Liu et al., 2023a; Zhao et al., 2024a; Bowman et al., 2015; Yoran et al., 2023; Kim et al., 2023; Li et al., 2024; Kumar et al., 2024; Ji et al., 2023; Zhao et al., 2025)). Commercial AI search engines directly deploy RAG to support answering arbitrary style queries. The information-retrieval module serves as the cornerstone of AI search engines, routinely returning unstructured, QA-oriented documents to supply in-context knowledge (Wang et al., 2024; Herzig et al., 2021; Liu et al., 2021; Zhao et al., 2024d). Consequently, **no existing retrieval solution offers a structured, query-adaptive schema for candidate comparison**, forcing downstream models to implicitly infer item attributes from vast, fragmented text. This limitation makes coherent recommendation list generation from hundreds of thousands of tokens prohibitively difficult. To bridge this critical gap, we propose the **Open-Domain Candidate Generation (OCG) Agent, the first agentic retrieval tool dedicated to can-**

didate recall.

Although recent structure-aware RAG variants (Zhao et al., 2024b; Han et al., 2025; Li et al., 2025a,b)—GraphRAG, SubgraphRAG, and StructRAG—prioritise graph-centric relational modelling, when transplanted to structured candidate retrieval, their graph modules add superfluous complexity and remain misaligned with task objectives, so substantial re-engineering is inevitable. By contrast, OCG-Agent is purpose-built for narrative-driven recommendation, offering a lean, task-aligned solution.

11 Conclusion

In this work, we formally define narrative-driven recommendation as a prevalent category of user queries across diverse applications and propose a tailored two-stage retrieve–rank paradigm to address its unique challenges. At the core of our framework is the Open-Domain Candidate Generation Agent (OCG-Agent), which autonomously produces structured and semantically aligned candidates, maximizing both the breadth and depth of information recall. Integrated with the RankGPT re-ranker, our OCG-RankGPT pipeline achieves significant gains in narrative recommendation performance compared with retrieve–then–read variants, standalone LLMs, commercial AI search engines, and deep-research approaches. This outperformance can be attributed to the wide-and-deep retrieval mechanism of the OCG-Agent and the reduced information loss and improved ranking accuracy afforded by our paradigm. Looking ahead, our framework remains agnostic to downstream rankers—inviting integration with advanced learning-to-rank models and agentic re-rankers—and positions OCG-Agent as a modular component in multi-agent collaboration systems, paving the way for more robust, context-rich narrative recommender applications.

Limitations

While our proposed retrieve–rank framework and OCG-Agent demonstrate substantial empirical and theoretical advantages, several limitations remain that merit discussion and future exploration. Our implementation adopts RankGPT (Sun et al.) as the re-ranking module within the retrieve–rank pipeline. Although RankGPT offers robust performance (typically achieving 60%–90% NDCG across various benchmarks), it is not necessarily the optimal choice. Recent advancements in agentic reasoning-based re-rankers (e.g., (Jin et al., 2025b,c)) present promising alternatives that could further enhance ranking accuracy. Nevertheless, our primary focus in this work is on the candidate-centric retrieval stage, which constitutes the central innovation of OCG-Agent. Future work could incorporate more sophisticated re-ranking models to further lift end-to-end performance. Besides, the evaluation of commercial AI search engines and deep-research systems was conducted in March 2025—a period during which deep research was beginning to be popular for answering any questions. As such, our reported findings represent a snapshot of system capabilities at a specific developmental phase and may not fully capture ongoing advancements in commercial deployments. Our experimental datasets are relatively small due to practical constraints. Specifically, the movie benchmark comprises 100 narrative queries instead of the full 296, primarily because executing OCG-Agent's candidate retrieval requires 30-60 minutes per query on average. Additionally, certain commercial systems impose API rate limits or very high usage costs that hinder large-scale testing. While we believe our sample size is sufficient to reveal consistent and statistically meaningful trends, expanding to larger datasets remains an important direction for strengthening the generalizability and robustness of our conclusions.

Ethics

This study adheres to rigorous ethical standards in both data collection and usage. The movie recommendation dataset is drawn from publicly available Reddit data, released under standard research-use licenses and curated in prior work (Eberhard et al., 2019, 2025). All data from this corpus are non-identifiable and freely accessible, ensuring compliance with ethical norms regarding user consent and privacy.The education dataset comprises 30 real-world narrative cases contributed by a local education consultancy. Each user involved in these cases provided informed consent for their narratives to be used in academic research. All personally identifiable information (e.g., names, birthdates, contact details) has been thoroughly removed or normalized, leaving only anonymized narrative queries that contain no privacy-sensitive content. Besides, the education dataset will not be publicly released to preserve institutional confidentiality.

Acknowledgments

This work was sponsored by the Australian Research Council under the Linkage Projects Grant LP210100129.

References

Jafar Afzali, Aleksander Mark Drzewiecki, and Krisztian Balog. 2021. Pointrec: A test collection for narrative-driven point of interest recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2478–2484, New York, NY, USA. Association for Computing Machinery.

Eslam Amer and Tamer Elboghdadly. 2024. The end of the search engine era and the rise of generative ai: A paradigm shift in information retrieval. In *2024 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 374–379.

Toine Bogers and Marijn Koolen. 2017. Defining and supporting narrative-driven recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 238–242, New York, NY, USA. Association for Computing Machinery.

Toine Bogers and Marijn Koolen. 2018. Narrative-driven recommendation as complex task. In *DIR 2018: 17th Dutch-Belgian Information Retrieval Workshop*, page 21.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2318–2335, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jiawei Chen, Xinyan Guan, Qianhao Yuan, Guozhao Mo, Weixiang Zhou, Yaojie Lu, Hongyu Lin, Ben He, Le Sun, and Xianpei Han. 2025. Consistentchat: Building skeleton-guided consistent dialogues for large language models from scratch. *Preprint*, arXiv:2506.03558.

Mengli Cheng, Yue Gao, Guoqiang Liu, Hongsheng Jin, and Xiaowen Zhang. 2022. Easyrec: An easy-to-use, extendable and efficient framework for building industrial recommendation systems. *ArXiv*, abs/2209.12766.

Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1126–1132, New York, NY, USA. Association for Computing Machinery.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Lukas Eberhard, Kristina Popova, Simon Walk, and Denis Helic. 2024. Computing recommendations from free-form text. *Expert Systems with Applications*, 236:121268.

Lukas Eberhard, Thorsten Ruprechter, and Denis Helic. 2025. Large language models as narrative-driven recommenders. In *Proceedings of the ACM on Web Conference 2025*, pages 4543–4561.

Lukas Eberhard, Simon Walk, and Denis Helic. 2020. Tell me what you want: Embedding narratives for movie recommendations. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, HT '20, page 301–306, New York, NY, USA. Association for Computing Machinery.

Lukas Eberhard, Simon Walk, Lisa Posch, and Denis Helic. 2019. Evaluating narrative-driven movie recommendations on reddit. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 1–11, New York, NY, USA. Association for Computing Machinery.

Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. 2023. A large language model enhanced conversational recommender system. *Preprint*, arXiv:2308.06212.

Marcos Fernández-Pichel, Juan C. Pichel, and David E. Losada. 2025. Evaluating search engines and large language models for answering health questions. *npj Digital Medicine*, 8(1):153.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey.

Yunfan Gao, Yun Xiong, Meng Wang, and Haofen Wang. 2024. Modular rag: Transforming rag systems into lego-like reconfigurable frameworks. *Preprint*, arXiv:2407.21059.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. 2025. Retrieval-augmented generation with graphs (graphrag). *Preprint*, arXiv:2501.00309.

Chao Hao, Shuai Wang, and Kaiwen Zhou. 2025. Uncertainty-aware gui agent: Adaptive perception through component recommendation and human-in-the-loop refinement. *Preprint*, arXiv:2508.04025.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian Mcauley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 720–730, New York, NY, USA. Association for Computing Machinery.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 512–519, Online. Association for Computational Linguistics.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers fornbsp;recommender systems. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part II*, page 364–381, Berlin, Heidelberg. Springer-Verlag.

Junjie Huang, Jizheng Chen, Jianghao Lin, Jiarui Qin, Ziming Feng, Weinan Zhang, and Yong Yu. 2024. A comprehensive survey on retrieval methods in recommender systems. *Preprint*, arXiv:2407.21022.

Junjie Huang, Jiarui Qin, Jianghao Lin, Ziming Feng, Weinan Zhang, and Yong Yu. 2025. Unleashing the potential of multi-channel fusion in retrieval for personalized recommendations. In *Proceedings of the ACM on Web Conference 2025*, pages 483–494.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, jiayi lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Guanglu Song, Peng Gao, Yu Liu, Chunyuan Li, and Hongsheng Li. 2025. Mmsearch: Unveiling the potential of large models as multi-modal search engines. In *The Thirteenth International Conference on Learning Representations*.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2025a. Long-context LLMs meet RAG: Overcoming challenges for long inputs in RAG. In *The Thirteenth International Conference on Learning Representations*.

Can Jin, Hongwu Peng, Anxiang Zhang, Nuo Chen, Jiahui Zhao, Xi Xie, Kuangzheng Li, Shuya Feng, Kai Zhong, Caiwen Ding, and Dimitris N. Metaxas. 2025b. Rankflow: A multi-role collaborative reranking workflow utilizing large language models. In *International Workshop on Resource-Efficient Learning for the Web, WWW 2025*.

Can Jin, Hongwu Peng, Shiyu Zhao, Zhenting Wang, Wujiang Xu, Ligong Han, Jiahui Zhao, Kai Zhong, Sanguthevar Rajasekaran, and Dimitris N. Metaxas. 2025c. Apeer: Automatic prompt engineering enhances large language model reranking. In *International Workshop on Resource-Efficient Learning for the Web, WWW 2025*.

Mingyu Jin, Weidi Luo, Sitao Cheng, Xinyi Wang, Wenyue Hua, Ruixiang Tang, William Yang Wang, and Yongfeng Zhang. 2024a. Disentangling memory and reasoning ability in large language models. *arXiv preprint arXiv:2411.13504*.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024b. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1830–1842.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in*

Natural Language Processing, pages 996–1009, Singapore. Association for Computational Linguistics.

Marijn Koolen, Toine Bogers, Maria GÃ¤de, Mark Hall, Iris Hendrickx, Hugo C. Huurdeman, Jaap Kamps, Mette Skov, Suzan Verberne, and David Walsh. 2016. Overview of the clef 2016 social book search lab. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 351–370.

Harsh Kumar, Ruiwei Xiao, Benjamin Lawson, Ilya Musabirov, Jiakai Shi, Xinyuan Wang, Huayin Luo, Joseph Jay Williams, Anna N. Rafferty, John Stamper, and Michael Liut. 2024. Supporting self-reflection at scale with large language models: Insights from randomized field experiments in classrooms. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, L@S '24, page 86–97, New York, NY, USA. Association for Computing Machinery.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Hanchung Lee. 2025. The differences between deep research, deep research, and deep research.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Mufei Li, Siqi Miao, and Pan Li. 2025a. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *International Conference on Learning Representations*.

Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2024. LLatrieval: LLM-verified retrieval for verifiable generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5453–5471, Mexico City, Mexico. Association for Computational Linguistics.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. 2025b. StructRAG: Boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization. In *The Thirteenth International Conference on Learning Representations*.

Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji-Rong Wen. 2023a. Reta-llm: A retrieval-augmented large language model toolkit. *arXiv preprint arXiv:2306.05212*.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023b. Is chatgpt a good recommender? a preliminary study. In *Proceedings of the 2023 GenRec Workshop at CIKM*. Accepted by CIKM 2023 GenRec Workshop.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023c. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (TACL)*. Accepted for publication.

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024. Once: Boosting content-based recommendation with both open- and closed-source large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 452–461, New York, NY, USA. Association for Computing Machinery.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip Yu. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuan Lu, Sifan Liu, Bochao Yin, Yongqi Li, Xinghao Chen, Hui Su, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2025. Multiconir: Towards multi-condition information retrieval. *Preprint*, arXiv:2503.08046.

Sebastian Lubos, Thi Ngoc Trang Tran, Alexander Felfernig, Seda Polat Erdeniz, and Viet-Man Le. 2024. Llm-generated explanations for recommender systems. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, UMAP Adjunct '24, page 276–285, New York, NY, USA. Association for Computing Machinery.

Xinbei Ma, Yeyun Gong, Pengcheng He, hai zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Sheshera Mysore, Andrew Mccallum, and Hamed Zamani. 2023. Large language model augmented narrative driven recommendations. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 777–783, New York, NY, USA. Association for Computing Machinery.

Ping Nie, Yujie Lu, Shengyu Zhang, Ming Zhao, Ruobing Xie, William Yang Wang, and Yi Ren. 2022. Mic: Model-agnostic integrated cross-channel recommender. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 3400–3409, New York, NY, USA. Association for Computing Machinery.

OpenAI, Josh Achiam, Steven Adler, and et.al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yunxiao Shi, Wujiang Xu, Zhang Zeqi, Xing Zi, Qiang Wu, and Min Xu. 2025a. PersonaX: A recommendation agent-oriented user modeling framework for long behavior sequence. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5764–5787, Vienna, Austria. Association for Computational Linguistics.

Yunxiao Shi, Xing Zi, Zijing Shi, Haimin Zhang, Qiang Wu, and Min Xu. 2024. Enhancing retrieval and managing retrieval: A four-module synergy for improved quality and efficiency in rag systems. In *ECAI 2024*, pages 2258–2265. IOS Press.

Zijing Shi, Meng Fang, and Ling Chen. 2025b. Monte carlo planning with large language model for text-based game agents. In *The Thirteenth International Conference on Learning Representations*.

Fabian Soto-Jiménez, Mateo Martínez-Velásquez, Janneth Chicaiza, Paola Vinueza-Naranjo, and Nadjet Bouayad-Agha. 2024. Rag-based question-answering systems for closed-domains: Development of a prototype for the pollution domain. In *Intelligent Systems and Applications*, pages 573–589, Cham. Springer Nature Switzerland.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? a study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.

Deep Search Team. 2024. Docling technical report. Technical report.

Dingmin Wang, Qiuyuan Huang, Matthew Jackson, and Jianfeng Gao. 2024. Retrieve what you need: A mutual learning framework for open-domain question answering. *Transactions of the Association for Computational Linguistics*, 12:247–263.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Wujiang Xu, Yunxiao Shi, Zujie Liang, Xuying Ning, Kai Mei, Kun Wang, Xi Zhu, Min Xu, and Yongfeng Zhang. 2025. iAgent: LLM agent as a shield between user and recommender systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18056–18084, Vienna, Austria. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *Preprint*, arXiv:2310.01558.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024a. Expel: Llm agents are experiential learners. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Jie Zhao, Lishuang Li, Wanting Ning, Jing Hao, Yuxiao Fei, and Jiayi Huang. 2024b. Multiple optimization with retrieval-augmented generation and fine-grained denoising for biomedical event causal relation extraction. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 4040–4043. IEEE.

Xinping Zhao, Baotian Hu, Yan Zhong, Shouzheng Huang, Zihao Zheng, Meng Wang, Haofen Wang, and Min Zhang. 2024c. Raserec: Retrieval-augmented sequential recommendation. *Preprint*, arXiv:2412.18378.

Xinping Zhao, Dongfang Li, Yan Zhong, Boren Hu, Yibin Chen, Baotian Hu, and Min Zhang. 2024d. SEER: Self-aligned evidence extraction for retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3027–3041, Miami, Florida, USA. Association for Computational Linguistics.

Xinping Zhao, Yan Zhong, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Dongfang Li, Baotian Hu, and Min Zhang. 2025. FunnelRAG: A coarse-to-fine progressive retrieval paradigm for RAG. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3029–3046, Albuquerque, New Mexico. Association for Computational Linguistics.

Xi Zhu, Yu Wang, Hang Gao, Wujiang Xu, Chen Wang, Zhiwei Liu, Kun Wang, Mingyu Jin, Linsey Pang, Qingsong Weng, Philip S. Yu, and Yongfeng Zhang. 2025a. Recommender systems meet large language model agents: A survey. *Foundations and Trends® in Privacy and Security*, 7(4):247–396.

Yaochen Zhu, Chao Wan, Harald Steck, Dawen Liang, Yesu Feng, Nathan Kallus, and Jundong Li. 2025b. Collaborative retrieval for large language model-based conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 3323–3334, New York, NY, USA. Association for Computing Machinery.

Yaochen Zhu, Liang Wu, Binchi Zhang, Song Wang, Qi Guo, Liangjie Hong, Luke Simon, and Jundong Li. 2024. Understanding and modeling job marketplace with pretrained language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 5143–5150.

Contents

Appendices

A Motivational Experiments

A.1 Setup

We select 30 queries specifically requesting movie recommendations, with the dataset sampled from prior research (Eberhard et al., 2019, 2024, 2025) available at [5]. GPT-4o-mini is employed as the representative LLMs as narrative-driven recommender due to its demonstrated effectiveness (Eberhard et al., 2025). For AI-driven search engines, we include ChatGPT-Search [6], Perplexity-Sonar [7], and Gemini-Search [8]. GPT-4o-mini can generate recommendation responses structured as JSON-formatted ranked lists. However, AI search engines can not directly produce valid JSON structures, thus requiring additional post-processing. To manage this issue, we utilize GPT-4o to extract and properly format these outputs. Prompt design uniformly adopts the chain-of-thought (CoT) (Wei et al., 2022), explicitly guiding LLMs to detail their reasoning processes step-by-step in free-text form. Furthermore, we instruct the models to include a concluding section explicitly delineating the final rankings, thereby enhancing JSON recognition accuracy and mitigating performance degradation associated with overly stringent formatting constraints (Tam et al., 2024).

For evaluation purposes, we employ standard top-10 ranking metrics, specifically precision, recall, and normalized discounted cumulative gain (NDCG) (Järvelin and Kekäläinen, 2002). Each query response is generated three times independently, with the final reported metrics representing averages across these iterations.

B Dataset Details

The Reddit MovieSuggestions dataset is the most widely adopted and canonical benchmark for narrative-driven recommendation (Bogers and Koolen, 2017; Eberhard et al., 2024, 2020, 2019, 2025). To move beyond this single-domain setting and assess real-world applicability, we additionally curated and evaluated a proprietary AusEdu-Narratives corpus comprising authentic overseas-study counselling cases collected by a professional consultancy. A partly relevant resource (Koolen et al., 2016) that could satisfy the evaluation needs of narrative-driven recommendation task is no longer accessible and contain sparse and noisy annotations.

Reddit MovieSuggestions. We use the benchmark released by Eberhard et al. (2019), which includes 1,483 movie recommendation threads from the r/MovieSuggestions subreddit. The final 20% of the data is held out as the test set. Each thread contains (i) a narrative-style query describing the user's preferences, and requirements, and (ii) community-suggested movie titles with up-vote counts. For evaluation, we randomly select 100 test queries. All recommended movies from each thread are merged and deduplicated. The final "oracle" ranking is based mainly on mention frequency while preserving the original up-vote order within each thread.

AusEdu-Narratives. We curated thirty real-world case studies from an Australian education consultancy [9]. Each case study comprises three core components—an academic profile (including native-scale GPA, IELTS score and notable awards), a personal background (country of origin, budget constraints and extracurricular interests) and the applicant's intent (desired discipline, intake term and preferred city)—together with the counsellor's ranked shortlist of appropriate programmes. We transform these discrete fields into a single coherent narrative queries so that our retrieval engine must jointly reason over both quantitative constraints (e.g. GPA 6.0/7.0) and qualitative preferences (e.g. "favors coastal locations"). All personally identifying information (names, birthdates and student identifiers) has been removed to guard against re-identification.

[5] https://doi.org/10.17605/osf.io/ma2bj
[6] https://openai.com/index/introducing-chatgpt-search/
[7] https://docs.perplexity.ai/home
[8] https://ai.google.dev/gemini-api/docs/grounding
[9] https://www.achieva-ai.com/home

C Baseline Details

To evaluate the effectiveness of our proposed approach, we compare it against three distinct paradigms.

LLM Direct This paradigm leverages the internal, parameterized knowledge of large language models (LLMs) for narrative recommendation tasks (Eberhard et al., 2025). Prior work has demonstrated that gpt-4o achieves state-of-the-art performance among both closed- and open-source LLMs (Eberhard et al., 2025). Therefore, we adopt (1) gpt-4o as a representative strong baseline. Moreover, motivated by emerging prompt-based models whose narrative recommendation performance has not been fully explored, we also include (2) deepseek-r1 (DeepSeek-AI, 2025) as an additional robust candidate.

AI Search Engine AI Search Engine methods employ a retrieval-augmented framework, wherein external web content is used to prompt LLMs in generating recommendations. This paradigm relies predominantly on externally sourced knowledge rather than the inherent parameters of the LLMs. For our evaluation, we incorporate three widely adopted AI Search Engines: (3) Perplexity, (4) GPT-AI Search, and (5) Gemini-AI Search.

Deep Research The Deep Research methods iteratively perform extensive searches and analyses to generate detailed reports (Lee, 2025). We mainly benchmark our approach against two representative deep research systems in whole dataset: (6) Perplexity Deep Research [10] and (7) Open Deep Research[11]. Due to the rapid emergence of Deep Research methods, most of which appeared after our study was largely completed, we were unable to conduct comprehensive evaluations of all Deep Research approaches on the full dataset. As a trade-off, we conducted supplementary evaluations on six other recently emerged Deep Research methods (both commercial and open-source) using the first five samples of Movie dataset, with detailed descriptions provided in Table 4.

Table 4: Comparison of Pioneer Deep Research Solutions on Movie Recommendation

Method	Version / Configuration	Precision@10	Recall@10	NDCG@10
OpenManus	Claude Sonnet 3.7 + Headless Web Browser	0.3750	0.1858	0.4121
Open Deep Research	GPT-4o + Serper Search	0.2750	0.1352	0.3412
OpenAI Deep Research	OpenAI o3 Reasoning	0.3250	0.1575	0.3555
Perplexity	Sonar Pro (in-platform web)	0.2500	0.1260	0.2744
Grok	Grok 3 (Deeper Research interface)	0.4250	0.2081	0.4630
Kimi	Kimi Researcher LLM + Kimi Search	0.1000	0.0451	0.1103
Qwen	Qwen-3 + Deep Thinking toolkit	0.1250	0.0675	0.1605
Gemini	Gemini 2.5 Pro + Deep Research retrieval	0.2750	0.1309	0.3341
OCG-RankGPT	RankGPT reranker (O3-mini backend)	0.3500	0.1726	0.4231

(We observed that Grok's Deeper Search achieved notably superior performance compared to other methods. However, upon careful investigation, we discovered that this product specifically targets Reddit's r/MovieSuggestions channel to extract user comments directly. This approach introduces ground truth data leakage. OpenManus demonstrated relatively strong performance, which we attribute to its comprehensive multi-channel search strategy across diverse platforms including BaiduSearchEngine, BingSearchEngine, DuckDuckGoSearchEngine, GoogleSearchEngine, and WebSearchEngine. In contrast, we only utilized Serper as the search engine. This performance difference reflects an engineering implementation gap rather than a fundamental method advantage.)

D Proof of Theorem 1

Assumption 1. There exists $\gamma \in [0,1]$ such that only γN of the retrieved items survive inherent information loss of LLM in handling long context. OCG-Agent can often achieve $\lambda \to 1$, such that OCG-Agent successfully recognizes and extracts every item in the $\mathcal{E}(q)$ yielding $\lim_{\lambda\to 1} C(q) = N$. There exists $\rho \in [0,1]$ satisfying $\Pr\big(\mathcal{E}(q) \supseteq \mathcal{I}^*_{\text{top}}(q)\big) \geq \rho$, where $\mathcal{I}^*_{\text{top}}(q) \subseteq \mathcal{I}$ is the true top-K relevant set. Leverage LLM for top-K ranking task yielding a accuracy of $\beta \in [0,1]$. Moreover, a sophisticated re-ranker achieves an accuracy of $\alpha \in [0,1]$ with $\alpha \geq \beta$.

[10] https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research
[11] https://github.com/langchain-ai/open_deep_research

Theorem 1. Under Assumption 1, the precision and recall of Retrieve-Read RAG paradigm and Retrieve-Rank satisfies

$$\mathbb{E}\big[\text{Precision@}K_{\text{RR}}\big] \geq \frac{N\rho\alpha}{K}, \tag{5}$$

$$\mathbb{E}\big[\text{Precision@}K_{\text{RAG}}\big] \geq \frac{\gamma N\rho\beta}{K}, \tag{6}$$

$$\mathbb{E}\big[\text{Recall@}K_{\text{RR}}\big] \geq \frac{N\rho\alpha}{|I^*(q)|}, \tag{7}$$

$$\mathbb{E}\big[\text{Recall@}K_{\text{RAG}}\big] \geq \frac{\gamma N\rho\beta}{|I^*(q)|}. \tag{8}$$

Proof. Limited by the LLMs' inherent limitations in handling long context article, it can only retain information from retrieved context at a fraction $\gamma \in (0, 1]$. Hence, out of the N candidates retrieved, only γN are successfully extracted. Moreover, at least a fraction $\rho \in (0, 1]$ of these γN candidates belong to the true top-K set $\mathcal{I}^*_{\text{top}}(q)$, i.e.

$$\big|\mathcal{I}^*_{\text{top}}(q) \cap C(q)\big| \;\geq\; \rho\gamma N.$$

Finally, if the LLM's ranking model achieves an accuracy of $\beta \in [0, 1]$ in placing relevant items within its Top-K output, then the number of true top-K items it correctly ranks is

$$\gamma N \;\times\; \rho \;\times\; \beta.$$

By definition of Precision@K and Recall@K, we thus obtain

$$\text{Precision@}K_{\text{RAG}} \;\geq\; \frac{\gamma N\rho\beta}{K}, \qquad \text{Recall@}K_{\text{RAG}} \;\geq\; \frac{\gamma N\rho\beta}{|I^*(q)|},$$

Then it is apparent that the proposed retrieve–rank paradigm delivers precision and recall as follows:

$$\mathbb{E}\big[\text{Precision@}K_{\text{RR}}\big] \;\geq\; \frac{N\rho\alpha}{K} > \frac{\gamma N\rho\beta}{K}, \qquad \mathbb{E}\big[\text{Recall@}K_{\text{RR}}\big] \;\geq\; \frac{N\rho\alpha}{|I^*(q)|} > \frac{\gamma N\rho\beta}{|I^*(q)|}.$$

□

E Case Study

Figure 4 provides a transparent depiction of both the retrieval and ranking processes through a representative case study, clarifying the real-world setting of the narrative recommendation task, detailing our OCG-RankGPT pipeline's workflow (including intermediate and end-to-end outputs), and offering a direct comparison against the ground truth.

F Prompt Templates

We illustrate representative prompt templates used in our study in Figure 5, Figure 6, Figure 7, and Figure 8. For the complete set of prompts, please refer to our publicly available code repository.

Case Study on Education Dataset

Narrative Query:
I want to study further. Can you help me? I am seeking guidance on pursuing a computer science related master degree in in Australia, starting in the second semester of 2025. I am an international student from China with a Bachelor of Engineering degree from Beijing University, a GPA of 3.7 on a scale of 88, and an IELTS score of 6.5. I am looking for recommendations on universities and programs that match my profile and preferences.

Ground Truth Recommendations:

1. Master of Computer Science at the University of Melbourne.
2. Master of Software Engineering at the University of Melbourne.
3. Master of Computer Science at the University of Sydney.
4. Master of Engineering Science at the University of New South Wales (UNSW Sydney).
5. Master of Engineering Science at the University of Queensland (UQ)
6. Master of Computer Science at the University of Queensland (UQ).
7. Master of Information Technology at the University of Technology Sydney (UTS).
8. Master of Engineering at the University of Technology Sydney (UTS).

Retrieved Candidate List:

1. Master of Science (Research) in Computing Sciences at UTS (University Ranking: 88, Admission Requirements: 6.5 overall with a writing score of 6.0 for IELTS, Major Componen: Previous qualifications must have a major computing component)
2. Master of Computer Science at The University of Melbourne (Scholarship Opportunities: Graduate Access Melbourne (GAM) for domestic students, General Admission Criteria: An undergraduate degree with a major in Computer Science with a WAM of at least 75%, English Language Requirements: IELTS 6.5 (with no band less than 6.0))
3. ...

Ranked Result:

1. Master of Software Engineering at the University of Melbourne.
2. Master of Information Technology at the University of Technology Sydney.
3. Master of Software Engineering at the University of Melbourne.
4. Master of Computer Science at The Australian National University.
5. Master of Engineering Science at the University of Queensland.
6. ...

Figure 4: Retrieved results and ranked result of OCG-RankGPT.

Abstract Data Type (ADT) Generation Prompt Template

Task Description:
Your task is to understand user's [Narrative Recommendation Query] and the analyzed [Personality Traits], then design appropriate Abstract Data Type (ADT) for the candidate item. You should consider what the type of things the candidate item is and what kinds of key attributes should be included.
Note that the attributes should be dynamically adjusted according to the user's concerns. For each attribute, it should be neither required or optional.
Narrative Recommendation Query:
{query}
Personality Traits:
{profile}
Analytical Steps:

1. Integrate the insights gained from the query with the personality indicators to identify the core attributes that the candidate item should possess.

2. For each identified attribute, furnish a detailed rationale that elucidates how the attribute aligns with the user's requirements while ensuring adaptability for dynamic adjustments.

Important Instructions:

1. All analyses must be strictly derived from the narrative query and the provided personality traits; no extraneous information should be incorporated.

2. Each inference and attribute selection must be supported by clear, logical evidence, ensuring the overall reasoning is both coherent and robust.

3. The design of the Abstract Data Type (ADT) should be responsive to the user's specific concerns, balancing the necessity of key attributes with the flexibility to accommodate optional requirements. The attribute "Name" is mandatory, whereas the attribute "Additional Information" is optional. The latter serves as a repository for supplementary descriptive details about the candidate that are not of primary importance.

Response Format:
Class Name: {classname}
Attributes:

    Name, required
    {attribute1}, {required/optional}
    {attribute2}, {required/optional}
    ...
    Additional information, optional

Figure 5: Prompt Template for Abstract Data Type (ADT) Generation: Design Candidate Attributes Tailored for User Concerns

Candidate Instance Extract Prompt Template

Task Description:
Your task is to extract relevant entities from the [Article] based on the given [ADT].
Abstract Data Type (ADT):
{ADT}
Article:
{article}
Important Instructions:

1. The primary objective is to extract instance object, the Abstract Data Type [ADT] is already defined and you should strictly follow.
2. Do not fabricate information—if an extracted instance object has incomplete attributes, keep them as NOT FOUND.
3. For the attribute 'Additional Information', it should be a JSON format containing supplementary descriptive details about the candidate. Or be an empty json.

Analytical Steps:

1. Read the [ADT] carefully and understand the defined data structure.
2. Read the [Article] then specific your founded instance object, list the Name attribute.
3. Write a section named 'Candidate List', followed by a json format answer.

Output Format:
You can articulate your thought process step by step in free text. However, at the end, you must generate a section titled 'Candidate List'. This section must be enclosed within triple backticks (```json) and (```). The 'Candidate List' should be formatted as JSON using the following structure:

````
```json
[
    {
        "attribute1": "{content}",
        "attribute2": "{content}",
        "...": "...",
        "Additional␣Information": {
            "xxx": "xxx",
            "..."
        }
    },
    "..."
]
```
````

Figure 6: Prompt Template for Candidate Extraction.

Single Query Generation Prompt Template

Task Description:
Your task is to generate just one query for searching, taking account for the [In Context Situation].
In Context Situation:
{in_context_situation}
Important Instructions:

1. Note: Just return single query, no else redundant words.

Analytical Steps:

1. Imagine the scenario in which the user is asking a question.
2. Simulate the user's thought process: What kind of query would they type into a search engine to easily find the information they are looking for?

Output Format:
You can think through the process step by step and ultimately generate a section titled 'Generated Query'. This section must be enclosed within triple backticks (```json ... ```). The 'Generated Query' should be formatted as JSON using the following structure. For example:

````
```json
{
    "query":"xxx"
}
```
````

Figure 7: Prompt Template for Single Query Generation for Targeted Search.

Incremental Attribute Completion Prompt Template

Task Description:
Your task is to complete the existing [Instance Object] based on the provided Abstract Data Type [ADT] and [Article].
Abstract Data Type (ADT):
{ADT}
Article:
{article}
Instance Object:
{candidate_item}
Important Instructions:

1. The primary objective is to complete existing [Instance Object] and do incremental information updation. The existing [Instance Object] has incomplete attributes value NOT FOUND. Your task is to fill these attributes if valuable information is provided in [Article].
2. The Abstract Data Type [ADT] is already defined and you should strictly follow.
3. Do not fabricate information.
4. For the attribute 'Additional Information', it should be a JSON format containing supplementary descriptive details about the candidate.

Analytical Steps:

1. Read the [ADT] carefully and understand the defined data structure.
2. Read the [Article] then specific your founded valuable information that can complete and do incremental updation to the existing [Instance Object].
3. Write a section named 'Completed Candidate', followed by a json format answer.

Output Format:
You can articulate your thought process step by step in free text. However, at the end, you must generate a section titled 'Completed Candidate'. This section must be enclosed within triple backticks (```json) and (```). The 'Completed Candidate' should be formatted as JSON aligning with [Instance Object], such as:

````
```json
{
    "Name": "{content}",
    "attribute1": "{content}",
    "attribute2": "{content}",
    ...,
    "Additional␣Information" : {
        "xxx" : "xxx",
        ...
    }
}
```
````

Figure 8: Prompt Template for Attribute Completion.